1971

Identification of system dynamics using multiple integrations.

Hansell, William Richard.

Monterey, California ; Naval Postgraduate School

http://hdl.handle.net/10945/15748

IDENTIFICATION OF SYSTEM DYNAMICS USING MULTIPLE INTEGRATIONS

William Richard Hansell

United States Naval Postgraduate School

THESIS

IDENTIFICATION OF SYSTEM DYNAMICS

USING MULTIPLE INTEGRATIONS

by

William Richard Hansell

Thesis Advisor: George J. Thaler

June 1971

Identification of System Dynamics

Using Multiple Integrations

by

William Richard Hansell
Ensign, United States Navy
B.E., Villanova University, 1970

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
June 1971

ABSTRACT

A practical method for identifying linear time invariant systems on the basis of arbitrary input-output records is reviewed and extended to handle the case where the system is not initially in the zero state. The method is implemented using a digital computer program composed of a numerical integration subroutine and a subroutine for solving overdetermined sets of linear algebraic equations. Several examples are presented to demonstrate the accuracy and present capabilities of the procedure.

TABLE OF CONTENTS

ACKNOWLEDGEMENT

The author wishes to express his appreciation to Doctor George J. Thaler of the Electrical Engineering Department of the Naval Postgraduate School for the guidance and helpful suggestions in the preparation of this work.

I. INTRODUCTION

A. THE IDENTIFICATION PROBLEM

In order to apply any of the modern techniques of control system design one must first have a mathematical model of the system to be controlled. The form of this model will depend on the design methods to be employed as well as on the physical characteristics of the system. Since most of the theory on the analysis and design of control systems is based either on the state space or transform representation of systems the vast majority of mathematical models will consist of either a set of state equations or a transfer function.

Once it has been decided what basic form the mathematical model should take the problem of determining the numerical values of the parameters arises. Parameter values can often be determined from the laws of physics and the data supplied by a manufacturer or obtained through testing. This is not always the case however. Occasionally the laws of physics become mathematically intractable or are not even applicable. Quite often the values of certain key parameters are not available. It is in these cases that the identification problem arises.

A common problem in engineering is that of determining the output of a system based on a knowledge of the system model, input, and initial conditions. The identification problem is similar to this but here the unknown quantity is

the system model. The input and output are assumed to be known. For the purposes of this thesis the identification problem can be stated as follows:

Given - a record of the input and output of a system over some finite period of time,

Find - a mathematical model and the numerical values of the model parameters in such a way that the model will accurately describe the behavior of the system.

It should be kept in mind that the problem of identifying a system solely on the basis of input and output data (the so called "black box" problem) is very rare in engineering. Even in the case where none of the model parameters are known one will more than likely have a fair idea of whether the system is linear or nonlinear, time varying or time invariant, what the order of magnitude of the dominant time constants is, and what types of inputs and outputs are to be expected. For this reason most engineering identification problems fall into the "grey box" category. This distinction may seem trivial but virtually all identification techniques rely heavily on knowledge of the characteristics and quantities mentioned above.

Although the control systems literature on system identification is quite vast there are no known identification schemes capable of handling all identification problems effectively. Choosing a method suitable for a given problem can become a formidable task. A paper summarizing

most of the common approaches to the problem of identifying lumped parameter systems has been published by Nieman, Fisher, and Seborg [1]. A good discussion of the industrial applications of various methods has been published by Eykhoff, *et al.* [2].

One common approach used in linear system identification is that of obtaining the impulse response of the system. Mishkin and Haddad [3] have developed a technique for finding the impulse response based on samples of the system output and its derivatives. A technique for estimating the impulse response on the basis of noisy input and output samples has been developed by Levin [4] and Kerr and Surber [5]. Turin [6] and Lichtenberger [7,8] have used a matched filter to obtain an identification. The use of a white noise or binary test signal and crosscorrelation has been suggested by Hill and McMurtry [9]. The noise limitations of the sample approximation, matched filter, and crosscorrelation identification techniques have been investigated by Lindenlaub and Cooper [10].

Another common approach to the identification problem is to determine the coëfficients of the differential or difference equation which describes the system. Kumar and Sridhar [11] have employed the method of quasilinearization with some success. Nagumo and Noda [12] have developed a learning approach to the problem. Bass [13] has developed a method which uses modulating functions and works well in the presence of noise. Astrom and Bohlin [14] have developed

a statistically optimal method of determining differential equation parameters known as the "maximum likelihood method." A similar method which is not optimal but is considerably simpler computationally has been developed by Peterka and Smuk [15,16]. It is known as the "prior knowledge fitting method." An algorithm for determining state variable models of sampled data systems has been proposed by Ho and Kalman [17]. The algorithm performs quite well in the presence of noise, and has been extended to continuously operating systems by Eldem [18].

Methods of identifying nonlinear and distributed parameter systems are usually limited to specific types of systems or to specific types of nonlinearities. This is undoubtedly due to the wide variety of nonlinearities encountered in physical systems and the difficulty of finding a model capable of characterizing them all. Shinbrot [19], Mowery [20], Fairman [21], and Bellman, *et al.* [22] have all approached the problem of identifying nonlinear systems by assuming a particular form of differential equation is capable of describing the system and then developing methods around the form of differential equation chosen. Another common approach to nonlinear system identification is that of representing a system by a suitable functional polynomial relating the input and output. Hsieh [23] uses this approach and a steepest descent algorithm to solve the identification problem. Similar approaches have been taken by Simpson [24], Bose [25,26], and Hubbell [27].

Identification methods vary widely with respect to how much must be known about the system before the method can be applied. Some identification techniques require that prior estimates of all system parameters be available. Many methods restrict the allowable system inputs to a family of testing functions such as steps or binary pulses. In general, the less that is known about a given system and the tighter the constraints on the kind of signals which may be applied as inputs the more difficult it is to find a method capable of accomplishing the identification.

B. OBJECTIVES OF THIS PAPER

This paper will present a study of an identification technique originally suggested by Diamessis [28]. It is designed to identify lumped linear time invariant systems but has been extended by Diamessis [29] and Wang [30] to handle certain types of nonlinearities. The technique requires a knowledge of the system input and output over some finite time interval as well as a rough estimate of the system order. The system input need not be restricted to a class of testing functions, it can be completely arbitrary.

Unlike some identification techniques which require the calculation of derivatives of the input and output, the technique to be presented requires only integrals of the input and output. The advantages of numerical integration over differentiation are well known. Since any zero mean noise component on the input or the output tends to be

attenuated greatly by the integration process the system identification can be more accurate than the raw data used to accomplish it.

The remainder of this thesis is divided into three major sections. In Chapter II the theoretical development of the identification technique is given. A method for identifying the initial conditions of the unknown system is also presented. Chapter III presents a method for implementing the techniques developed in Chapter II. Particular attention is given to the choice of numerical methods and to efficient programming techniques. Several examples are presented to demonstrate the capabilities of the method. In Chapter IV several recommendations for further study are made. Conclusions concerning the accuracy and present limitations of the technique under consideration are also discussed. Following the conclusions a complete listing of all computer programs used in the thesis is given.

II. IDENTIFICATION BY MULTIPLE INTEGRATIONS

A. GENERAL APPROACH

The development which follows is similar to the development given by Diamessis [28] in 1965. There are a few notable differences however. The development given by Diamessis is restricted to the case where all initial conditions are zero. This is a rather serious restriction since it may be difficult or impossible to find a point where the system is in the zero state if the systems operation is not to be disturbed. Zero initial conditions will not be assumed in the development which follows. A method for solving for the unknown initial conditions will be presented. Diamessis proposed the formulation of a uniquely determined set of linear algebraic equations with the model parameters as unknowns. This development will make use of overdetermined sets of linear algebraic equations with the model parameters and initial conditions as unknowns. The overdetermined set of equations will then be solved using the method of least squares. It will be shown that this results in a more accurate identification when the accuracy of the available data is limited and the order of the system is unknown.

Any single input, single output, lumped parameter, linear, time-invariant system can be described by a linear ordinary differential equation with constant coefficients.

The basic form of this equation is given in Equation (1) along with a set of initial conditions.

$$\frac{d^n y(t)}{dt^n} + a_{n-1} \frac{d^{n-1} y(t)}{dt^{n-1}} + \cdots\cdots\cdots + a_o y(t) \qquad (1)$$

$$= b_m \frac{d^m u(t)}{dt^m} + \cdots\cdots\cdots + b_o u(t)$$

with initial conditions;

$$\frac{d^{n-1} y(0)}{dt^{n-1}} = y_o{}^{n-1} \qquad \frac{d^{m-1} u(0)}{dt^{m-1}} = u_o{}^{m-1}$$

$$\vdots \qquad\qquad \vdots$$

$$y(0) = y_o \qquad u(0) = u_o$$

where;

u(t) = system input

y(t) = system output

The identification problem to be treated here consists of determining n, m, and the various coefficients of the differential equation on the basis of input and output records taken over some arbitrary time interval. The initial conditions will be assumed to be unknown.

Taking the Laplace transform of Equation (1) yields Equation (2).

$$s^n Y(s) + a_{n-1} s^{n-1} Y(s) + \cdots\cdots\cdots + a_o Y(s) \qquad (2)$$

$$= \quad b_m s^m U(s) + \cdots\cdots\cdots + b_o U(s)$$

$$+ \quad g_{n-1} s^{n-1} + \cdots\cdots\cdots + g_o .$$

The g_i coefficients account for the contributions of the initial conditions. Dividing Equation (2) by s^{n+1} is equivalent to integrating n+1 times in the time domain.

$$\frac{Y(s)}{s} + a_{n-1}\frac{Y(s)}{s^2} + \cdots\cdots\cdots + a_o\frac{Y(s)}{s^{n+1}} \tag{3}$$

$$= b_m \frac{U(s)}{s^{n-m+1}} + \cdots\cdots\cdots + b_o \frac{U(s)}{s^{n+1}}$$

$$+ g_{n-1}\frac{1}{s^2} + \cdots\cdots\cdots + g_o \frac{1}{s^{n+1}} .$$

Taking the inverse transform of Equation (3) results in Equation (4).

$$\int_0^{t_k} y(t)dt + a_{n-1}\int_0^{t_k}\int_0^{t_k} y(t)dt^2 + \tag{4}$$

$$\cdots + a_o \int_0^{t_k} \cdots n+1 \cdots \int_0^{t_k} y(t)dt^{n+1}$$

$$= b_m \int_0^{t_k} \cdots n-m+1 \cdots \int_0^{t_k} u(t)dt^{n-m+1} + \cdots$$

$$\cdots + b_o \int_0^{t_k} \cdots n+1 \cdots \int_0^{t_k} u(t)dt^{n+1}$$

$$+ g_{n-1}\int_0^{t_k}\int_0^{t_k} dt^2 + \cdots + g_o \int_0^{t_k} \cdots n+1 \cdots \int_0^{t_k} dt^{n+1} .$$

Since the system is time invariant nothing has been lost by setting the lower limit on the integrals equal to zero.

Rearranging terms in Equation (4) and placing all terms which depend on u(t), y(t), or t in brackets results in Equation (5).

$$a_o \int_o^{t_k} ..n{+}1.. \int_o^{t_k} y(t)dt^{n+1} + \cdots + a_{n-1} \int_o^{t_k} \int_o^{t_k} y(t)dt^2$$
$$+ b_o \int_o^{t_k} ..n{+}1.. \int_o^{t_k} u(t)dt^{n+1} + \cdots$$
$$\cdots + b_m \int_o^{t_k} ..n{-}m{+}1.. \int_o^{t_k} u(t)dt^{n-m+1} + g_o \int_o^{t_k} ..n{+}1.. \int_o^{t_k} dt^{n+1}$$
$$+ \cdots + g_{n-1} \int_o^{t_k} \int_o^{t_k} dt^2 = - \int_o^{t_k} y(t)dt \quad . \tag{5}$$

Since records of the input and output are assumed to be known a linear algebraic equation with the system parameters and initial condition terms as unknowns can be formulated by performing the indicated multiple integrations from zero to some time t_k. A set of 2n+m+1 equations can be obtained by letting t_k take on 2n+m+1 different values. Assuming that the equations are linearly independent it will now be possible to solve for the n+m+1 differential equation coefficients and the n initial condition terms.

So far nothing has been said of how n and m are determined. Theoretically it should be possible to use any

n' and m' greater than the actual order of the system under study. If the order of the system is n with m input coefficients then one would expect the following:

$$a_i = 0 \quad \text{for } (0 \leq i \leq n'-n-1)$$

$$b_i = 0 \quad \text{for } (0 \leq i \leq m'-m-1)$$

$$\text{with } n'>n \quad \text{and} \quad m'>m.$$

The model should essentially reduce itself to the right order by setting nonessential terms equal to zero.

Unfortunately the situation is not quite this simple. Due to the finite accuracy of all experimental data the linear equations will not have an exact solution. For certain types of inputs the linear equations will not even be linearly independent. These problems can be overcome to some extent by formulating more than 2n+m+1 equations which are required. The overdetermined set can then be solved using the method of least squares. If the limited accuracy of the experimental data can be attributed to round off errors then integrating the data over a time interval much greater than the sampling interval should remove much of the uncertainty in the linear equation coefficients. This is because the roundoff process can be modeled as a zero mean white noise process. The integral of the noise will approach zero as the period of integration increases.

Even the measures mentioned above will not solve the problem completely however. Due to the finite precision used to represent numbers in a computer and the iterative

nature of the numerical methods used to solve overdetermined sets of linear equations it is impossible to obtain zero as a solution for any parameter. If the parameter should be zero the numerical method will return a very small but nonzero value. Although this will result in an incorrect estimate of the system order the error will not be serious in most cases. This is because the small coefficients of the terms which should nonexistent will make their effect negligible. Examples in Chapter III will demonstrate this point.

Although the development in this section has assumed a differential equation model of the system the same results could have been obtained if a transfer function or state variable model had been assumed. A simple rearrangement of terms in Equation (2) results in Equation (6).

$$\frac{Y(s)}{U(s)} = \frac{b_m s^m + \cdots\cdots\cdots + b_1 s + b_o}{s^n + a_{n-1} s^{n-1} + \cdots\cdots + a_1 s + a_o} \quad . \tag{6}$$

By defining a few new terms, Equation (7) results.

$$\frac{Y(s)}{U(s)} = \frac{K\ (s^m + c_{m-1} s^{m-1} + \cdots\cdots + c_1 s + c_o}{s^n + a_{n-1} s^{n-1} + \cdots\cdots + a_1 s + a_o} \tag{7}$$

where $K = b_m$

$c_i = b_i \,/\, b_m \quad$ for $0 \leq i \leq m$.

A set of state equations may be formulated in a similar fashion. One convenient state variable representation is given in Equation (8). It is based on the phase variable form of system representation.

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \cdot \\ \cdot \\ \cdot \\ \dot{x}_n \end{bmatrix} = \left[\begin{array}{c|cccc} 0 & 1 & & & \\ 0 & 0 & \ddots & & \\ \cdot & & & \ddots & \\ \cdot & & & & \\ 0 & 0 & & & 1 \\ \hline -a_o & -a_1 & \cdots & & -a_n \end{array}\right] \times \begin{bmatrix} x_1 \\ x_2 \\ \cdot \\ \cdot \\ \cdot \\ x_n \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \cdot \\ \cdot \\ \cdot \\ K \end{bmatrix} [\ u\ ] \tag{8}$$

$$y = \begin{bmatrix} c_o & c_1 & \cdots & c_m & 0 & \cdots & 0 \end{bmatrix} \times \begin{bmatrix} x_1 \\ x_2 \\ \cdot \\ \cdot \\ x_n \end{bmatrix}$$

Note the simple correspondence between the terms in the transfer function and the terms in the state equations.

B. IDENTIFYING INITIAL CONDITIONS

In many identification problems it is desirable to compare the system model with the actual system by exciting the system model with the same input data that was used in the identification. By comparing the output of the model with the output of the system a rough idea of the accuracy of the model can be obtained. This will not be possible however unless the initial conditions at the beginning of the input-output record are all known.

From Equation (5) it can be seen that when the linear algebraic equations are solved for the unknown model parameters

the g_i initial condition terms are also found. By taking the Laplace Transform of Equation (1) and specifically writing in the contributions of the individual initial conditions a relationship between the g_i terms and the initial conditions can be found.

$$
\begin{bmatrix} g_o \\ g_1 \\ \cdot \\ \cdot \\ \cdot \\ g_{n-2} \\ g_{n-1} \end{bmatrix}
=
\begin{bmatrix}
a_1 & a_2 & \cdots & a_{n-1} & 1 \\
a_2 & a_3 & & 1 & \\
\cdot & \cdot & \iddots & & \\
\cdot & \cdot & & 0 & \\
\cdot & \cdot & & & \\
a_{n-1} & 1 & & & \\
1 & & & &
\end{bmatrix}
\times
\begin{bmatrix} y_o \\ y_o^{1} \\ \cdot \\ \cdot \\ \cdot \\ y_o^{n-2} \\ y_o^{n-1} \end{bmatrix}
\tag{9}
$$

$$
-K \times
\left[\begin{array}{ccccc|ccc}
c_1 & c_2 & \cdots & c_{m-1} & 1 & 0 & \cdots & 0 \\
c_2 & c_3 & \cdot & 1 & & & & \\
\cdot & \cdot & \iddots & & & & & \\
\cdot & \cdot & & 0 & & & 0 & \\
c_{m-1} & 1 & & & & & & \\
1 & & & & & 0 & \cdots & 0 \\
\hline
0 & & & & & 0 & \cdots & 0 \\
\cdot & & 0 & & & & & \cdot \\
\cdot & & & & & & & \cdot \\
0 & & & & & 0 & \cdots & 0
\end{array}\right]
\times
\left[\begin{array}{c}
u_o \\ u_o^{1} \\ \cdot \\ \cdot \\ u_o^{m-2} \\ u_o^{m-1} \\ \hline 0 \\ \cdot \\ \cdot \\ 0
\end{array}\right]
$$

Unfortunately Equation (9) requires the knowledge of m-1 derivatives of the input function u. It may be necessary to calculate m-1 derivatives of the input using numerical techniques. This could cause the model and system output to differ slightly at the beginning of the output record but as the natural response dies out the records should converge. It may be possible to avoid this difficulty in many cases by choosing the beginning of the input-output record at a point where the input is relatively constant.

C. SIMPLIFICATIONS WITH ZERO INITIAL CONDITIONS

In many problems it will be possible to exercise complete control over the input to the system under study. If it is possible to obtain an input-output record beginning when the system is in the zero state it will be possible to simplify the identification procedure. Since the g_i terms will all be zero if the system is in the zero state they need not be included in the formulation of the linear algebraic equations. This will reduce the number of unknowns from 2n+m+1 to n+m+1.

It should be noted that additional information can often be incorporated into the identification procedure in order to simplify the problem. For example if the steady state gain constant were known the number of unknowns could be reduced by one. It is usually a simple matter to tell whether a system has poles or zeroes at the origin. This

piece of information can easily be used to simplify the identification procedure still further. As a general rule, the fewer the unknowns the more accurate the identification.

III. IMPLEMENTATION

A. NUMERICAL METHODS

The identification technique presented in Chapter II can be broken into two basic steps. The first step consists of performing the multiple integrations of the input and output and forming the overdetermined set of linear algebraic equations. The second step consists of solving the linear equations for the unknown model parameters. It is a distinct advantage of the identification technique under study that each of these steps can be carried out by subprograms that are readily available in virtually all modern computer centers.

Step one can be handled by any numerical integration subroutine. Although there are quite a few highly sophisticated numerical integration procedures available, trapezoidal integration will give better results in most applications. There are several reasons why this is true. First of all, most of the more complex integration techniques perform poorly when the function being integrated is discontinuous. Since control system inputs are often discontinuous and since such discontinuities are quite desirable from an identification standpoint, complex integration techniques are usually undesirable. Even when the functions to be integrated are continuous the slight improvement in accuracy offered by more advanced methods is not enough to justify the tremendous increase in computational load associated with their use.

Step two, the solution of the set of overdetermined linear equations, is a classical problem in several fields of mathematics and engineering. Unfortunately most of the classical techniques for solving such problems are not practical. They tend to magnify the errors introduced by the finite precision of the computer to the point where the solution is meaningless. Fortunately several modern methods are available that display more acceptable behavior. The method used in this paper was developed by Golub [31] in 1965. The basic approach is to triangularize the coefficient matrix by performing a Choleski decomposition. The decomposition is accomplished by applying repeated Householder transformations [32]. Once the coefficient matrix has been triangularized the unknowns can be obtained by back substitution. The method is quite stable numerically and is capable of handling illconditioned coefficient matrices.

B. CHARACTERISTICS OF GOOD INPUT-OUTPUT RECORDS

The accuracy with which a system can be identified is strongly dependent on the input-output record used in the identification. Since parameters are identified on the basis of their effect on the output it will be impossible to identify a parameter unless its effect is measureable. If the input to a system has a frequency spectrum that is more or less uniform over the frequency range of interest the identification will probably be very good. If the

frequency spectrum of the input is confined to a narrow band of frequencies the identification will probably be very bad. It is well known that signals with sharp discontinuities have a broader bandwidth than slowly varying continuous signals. For this reason input-output records displaying discontinuities and rapid time variations should be chosen.

If step or ramp inputs are used in the identification the value of the initial conditions will have to be known and incorporated into the set of linear equations. Since the initial conditions will usually be zero when these types of inputs are used this should not cause any difficulties. The reason for this difficulty lies in the nature of the initial condition coefficient terms. The integral coefficients of these terms are steps, ramps, and higher order polynomials in time. If the input is a step or a ramp the coefficients of several model parameters will also be steps, ramps, and higher order polynomials in time. There will therefore be a direct relationship between model parameter coefficients and initial condition term coefficients. This will result in the linear equations having an infinite number of solutions due to the linear dependence between all the individual equations in the set. Step and ramp inputs must therefore be avoided when the system initial conditions are unknown.

Since analog system data will have to be converted to digital form a suitable sampling interval will have to be

chosen. Experimental results have shown that a sampling rate ten to one hundred times shorter than the shortest system time constant works quite well. Lower sampling rates may introduce inaccuracies in the location of high frequency poles and zeroes.

C. PROGRAM DESCRIPTION

In order to evaluate experimentally the characteristics of the identification procedure under study a set of digital computer programs was developed. The main identification program is a direct implementation of the procedure developed in Chapter II. Subroutine SYSTEM is a simulation program that was written to generate input-output data for the main program to process.

In the beginning of the identification program several important parameters are defined. NP and NZ are rough estimates of the number of poles and the number of zeroes in the system to be identified. KPTMAX is the number of sample points available in the input-output record. Each sample point will consist of the time (T), the input amplitude (R), and the output amplitude (C). IPTS is the number of sample points that will separate successive linear equations. In other words, every time the total number of points read in is a multiple of IPTS a linear equation will be generated. The total number of linear equations that will be generated is equal to KPTMAX/IPTS. When all of the linear equations have been formed subroutine DLLSQ is called. This

subroutine is an implementation of the Golub algorithm for solving overdetermined sets of linear equations.

Subroutine DLLSQ returns the values of the system model parameters and initial condition parameters of Equation (2). In order to find the poles and zeroes of the system the output of DLLSQ is fed into RTPLSB. RTPLSB is a polynomial root finder which uses a combination of the Newton-Raphson and Bairstow methods to find the poles and zeroes of the system.

The remainder of the main identification program is devoted to output. The results of the identification are given in both transfer function and state variable form. The state variable form is referenced to the format used in Equation (8).

Subroutine SYSTEM reads in the transfer function of a system and computes the system output based on a set of arbitrary initial conditions and an arbitrary input waveform. Each time subroutine SYSTEM is called by the identification program it returns three numbers, the time T, the input waveform amplitude R, and the output waveform amplitude C. Subroutine SYSTEM prints out the transfer function and state variable representation of the system it is simulating so that the accuracy of the identification can be determined.

Input-output recrods obtained from physical systems are rarely accurate to more than three or four significant digits. The data generated by subroutine SYSTEM is therefore

rounded off by subroutine ROUND before being passed to the identification program. The number of significant digits in the data returned to the identification program may be varied by changing the value of the parameter NA in the simulation subroutine.

D. EXAMPLES

The following examples demonstrate how the accuracy of an identification depends on the accuracy of the input-output record, the sampling period, and the input waveform. They also show how the order of the system can be determined from a trial identification run using an estimated order greater than the actual order of the system.

Example one illustrates the relationship between the accuracy of the input-output record data and the resulting identification. In order to minimize the effect of other factors all initial conditions were set equal to zero, a step input was used, and n' and m' were set equal to n and m. Example one demonstrates the fact that there is a direct (almost linear) relationship between the accuracy of input-output data and the accuracy of the identification. Note that even when the input-output record contained only two significant digits the identification of the system poles and zeroes was within about three percent of their exact values.

Example two illustrates the relationship between the sampling period used with the input-output records and the

the accuracy of the identification, input records containing discontinuities or rapid time variations should be chosen.

Examples four through ten demonstrate what happens when n' and m' are greater than n and m. In each example an identification is performed using an n' and m' greater than n and m. Using the information obtained from this identification a new value of n' and m' is determined. These new estimates are then used to perform a second identification.

In examples four and five the error in the estimate of m' was made larger than the error in the estimate of n'. As a result of the relative values of these two errors all the excess poles cancelled with excess zeroes but since there were more excess zeroes than excess poles some excess zeroes remained. Note however that for frequencies lower than the sampling rate the excess zeroes have little or no effect on the behavior of the identified system. Experiments have shown that excess zeroes that do not cancel with excess poles are always of a frequency comparable to or higher than the sampling rate. Using this principle and estimating new values for n' and m' results in an identification which has the correct number of poles and zeroes and is more accurate than the first identification.

In examples six, seven, and eight the error in the estimate of n' was equal to the error in the estimate of m'. As a result, all excess poles and zeroes cancelled

with each other leaving a system of the correct order. Note that by repeating the identification with the correct value of n' and m' it was possible to improve the accuracy of the identification.

In examples nine and ten the error in the estimate of n' was made greater than the error in the estimate of m'. As a result, all excess zeroes cancelled with excess poles but some excess poles remained. Note that for frequencies lower than the sampling rate the excess poles have negligible effect. Experiments have shown that excess poles that do not cancel with excess zeroes are almost always of a frequency comparable to or greater than the sampling rate. Using this principle and estimating new values for n' and m' resulted in a correct identification of the simulated systems.

Experiments have shown that identifications involving uncancelled excess zeroes are generally more accurate than identifications involving uncancelled excess poles. For this reason it is best to set m' close to n' when identifying an unknown system.

Using the experimental findings described above a simple procedure for determining n and m can be formulated. First, guess an n' which is greater than the order of the system to be identified. This should not be too difficult in most engineering identification problems. Let m' be equal to n' or m'-1. This will guarantee complete cancellation of all excess poles and zeroes or partial cancellation

of excess poles and zeroes with excess zeroes remaining. After making a trial identification with the values of n' and m' mentioned above, estimate new values for n' and m' based on the reasoning in the examples. The new values of n' and m' should now be correct. By performing the identification with these values of n' and m' a good identification should result.

EXAMPLE 1.

SYSTEM TO BE IDENTIFIED

SYSTEM TRANSFER FUNCTION

POLES	REAL	IMAGINARY	
1	-1.0000000000	0.0	J
2	-10.0000000000	0.0	J
3	-100.0000000000	0.0	J

ZEROES	REAL	IMAGINARY	
1	-3.0000000000	0.0	J
2	-30.0000000000	0.0	J

GAIN CONSTANT = 10.0000000000

SYSTEM STATE VARIABLES (PHASE FORM)

A VECTOR

1	1000.0000000000
2	1110.0000000000
3	111.0000000000

B VECTOR

3	10.0000000000

C VECTOR

1	90.0000000000
2	33.0000000000
3	1.0000000000

IDENTIFICATION OF UNKNOWN SYSTEM

SYSTEM TRANSFER FUNCTION

POLES	REAL	IMAGINARY	
1	-100.0001690225	0.0	J
2	-0.9999970711	0.0	J
3	-9.9999884241	0.0	J

ZEROES	REAL	IMAGINARY	
1	-2.9999937654	0.0	J
2	-30.0000138488	0.0	J

GAIN CONSTANT = 10.0000038147

SYSTEM STATE VARIABLES (PHASE FORM)

A VECTOR

1	999.9976037596
2	1110.0003679024
3	111.0001545177

B VECTOR

3	10.0000038147

C VECTOR

1	89.9998545091
2	33.0000076143
3	1.0000000000

PROGRAM PARAMETERS

RECORD LEGNTH	=	0.60000 SEC
SAMPLING PERIOD	=	0.20000 MSC
EQUATIONS FORMED	=	120
PRECISION OF DATA	=	8 DECIMAL PLACES
INPUT FUNCTION	=	UNIT STEP

IDENTIFICATION OF UNKNOWN SYSTEM

SYSTEM TRANSFER FUNCTION

POLES	REAL	IMAGINARY	
1	-100.0000291125	0.0	J
2	-1.0000011522	0.0	J
3	-10.0000016550	0.0	J

ZEROES	REAL	IMAGINARY	
1	-3.0000028353	0.0	J
2	-30.0000250480	0.0	J

GAIN CONSTANT = 9.9999933243

SYSTEM STATE VARIABLES (PHASE FORM)

A VECTOR

1	1000.0016088045
2	1110.0006141285
3	111.0000319197

B VECTOR

3	9.9999933243

C VECTOR

1	90.0001602028
2	33.0000278833
3	1.0000000000

PROGRAM PARAMETERS

RECORD LEGNTH	=	0.60000 SEC
SAMPLING PERIOD	=	0.20000 MSC
EQUATIONS FORMED	=	120
PRECISION OF DATA	=	7 DECIMAL PLACES
INPUT FUNCTION	=	UNIT STEP

IDENTIFICATION OF UNKNOWN SYSTEM

SYSTEM TRANSFER FUNCTION

POLES	REAL	IMAGINARY	
1	-100.0000683416	0.0	J
2	-1.0000118079	0.0	J
3	-10.0000329083	0.0	J

ZEROES	REAL	IMAGINARY	
1	-3.0000272949	0.0	J
2	-30.0000668753	0.0	J

GAIN CONSTANT = 9.9999847412

SYSTEM STATE VARIABLES (PHASE FORM)

A VECTOR

1	1000.0157822215
2	1110.0053743721
3	111.0001130579

B VECTOR

3	9.9999847412

C VECTOR

1	90.0010194754
2	33.0000941702
3	1.0000000000

PROGRAM PARAMETERS

RECORD LEGNTH	=	0.60000 SEC
SAMPLING PERIOD	=	0.20000 MSC
EQUATIONS FORMED	=	120
PRECISION OF DATA	=	6 DECIMAL PLACES
INPUT FUNCTION	=	UNIT STEP

IDENTIFICATION OF UNKNOWN SYSTEM

SYSTEM TRANSFER FUNCTION

POLES	REAL	IMAGINARY	
1	-99.9968491528	0.0	J
2	-1.0000017165	0.0	J
3	-9.9998390264	0.0	J

ZEROES	REAL	IMAGINARY	
1	-2.9999754422	0.0	J
2	-29.9992863605	0.0	J

GAIN CONSTANT = 9.9998254776

SYSTEM STATE VARIABLES (PHASE FORM)

A VECTOR

1	999.9541111369
2	1109.9492716708
3	110.9966898958

B VECTOR

3	9.9998254776

C VECTOR

1	89.9971223661
2	32.9992618027
3	1.0000000000

PROGRAM PARAMETERS

RECORD LEGNTH	=	0.60000 SEC
SAMPLING PERIOD	=	0.20000 MSC
EQUATIONS FORMED	=	120
PRECISION OF DATA	=	5 DECIMAL PLACES
INPUT FUNCTION	=	UNIT STEP

IDENTIFICATION OF UNKNOWN SYSTEM

SYSTEM TRANSFER FUNCTION

POLES	REAL	IMAGINARY	
1	-99.9927835443	0.0	J
2	-0.9991304951	0.0	J
3	-9.9980805636	0.0	J

ZEROES	REAL	IMAGINARY	
1	-2.9981092090	0.0	J
2	-29.9981888298	0.0	J

GAIN CONSTANT = 9.9994869232

SYSTEM STATE VARIABLES (PHASE FORM)

A VECTOR

1	998.8666303721
2	1109.6311321656
3	110.9899946030

B VECTOR

3	9.9994869232

C VECTOR

1	89.9378461845
2	32.9962980388
3	1.0000000000

PROGRAM PARAMETERS

RECORD LEGNTH	=	0.60000 SEC
SAMPLING PERIOD	=	0.20000 MSC
EQUATIONS FORMED	=	120
PRECISION OF DATA	=	4 DECIMAL PLACES
INPUT FUNCTION	=	UNIT STEP

IDENTIFICATION OF UNKNOWN SYSTEM

SYSTEM TRANSFER FUNCTION

POLES	REAL	IMAGINARY	
1	-99.9937910594	0.0	J
2	-0.9996622783	0.0	J
3	-10.0015833055	0.0	J

ZEROES	REAL	IMAGINARY	
1	-2.9996507024	0.0	J
2	-30.0061026888	0.0	J

GAIN CONSTANT = 9.9986057281

SYSTEM STATE VARIABLES (PHASE FORM)

A VECTOR	
1	999.7584771470
2	1110.0544578539
3	110.9950366432

B VECTOR	
3	9.9986057281

C VECTOR	
1	90.0078270058
2	33.0057533912
3	1.0000000000

PROGRAM PARAMETERS

RECORD LEGNTH	=	0.60000 SEC
SAMPLING PERIOD	=	0.20000 MSC
EQUATIONS FORMED	=	120
PRECISION OF DATA	=	3 DECIMAL PLACES
INPUT FUNCTION	=	UNIT STEP

IDENTIFICATION OF UNKNOWN SYSTEM

SYSTEM TRANSFER FUNCTION

POLES	REAL	IMAGINARY	
1	-98.8004578528	0.0	J
2	-0.9763637189	0.0	J
3	-9.8862584052	0.0	J

ZEROES	REAL	IMAGINARY	
1	-2.9406610975	0.0	J
2	-29.6497245014	0.0	J

GAIN CONSTANT = 9.9451894760

SYSTEM STATE VARIABLES (PHASE FORM)

A VECTOR	
1	953.6797208413
2	1082.8846233636
3	109.6630799769

B VECTOR	
3	9.9451894760

C VECTOR	
1	87.1897913926
2	32.5903855989
3	1.0000000000

PROGRAM PARAMETERS

RECORD LEGNTH	=	0.60000 SEC
SAMPLING PERIOD	=	0.20000 MSC
EQUATIONS FORMED	=	120
PRECISION OF DATA	=	2 DECIMAL PLACES
INPUT FUNCTION	=	UNIT STEP

EXAMPLE 2.

SYSTEM TO BE IDENTIFIED

SYSTEM TRANSFER FUNCTION

POLES	REAL	IMAGINARY	
1	-1.0000000000	0.0	J
2	-10.0000000000	0.0	J
3	-100.0000000000	0.0	J

ZEROES	REAL	IMAGINARY	
1	-3.0000000000	0.0	J
2	-30.0000000000	0.0	J

GAIN CONSTANT = 10.0000000000

SYSTEM STATE VARIABLES (PHASE FORM)

A VECTOR

1	1000.0000000000
2	1110.0000000000
3	111.0000000000

B VECTOR

3	10.0000000000

C VECTOR

1	90.0000000000
2	33.0000000000
3	1.0000000000

IDENTIFICATION OF UNKNOWN SYSTEM

SYSTEM TRANSFER FUNCTION

POLES	REAL	IMAGINARY	
1	-101.0610717819	0.0	J
2	-1.0000331312	0.0	J
3	-10.0017011716	0.0	J

ZEROES	REAL	IMAGINARY	
1	-3.0002183624	0.0	J
2	-30.0553440030	0.0	J

GAIN CONSTANT = 10.0887174606

SYSTEM STATE VARIABLES (PHASE FORM)

A VECTOR

1	1010.8161284948
2	1121.8490926385
3	112.0628060847

B VECTOR

3	10.0887174606

C VECTOR

1	90.1725949663
2	33.0555623654
3	1.0000000000

PROGRAM PARAMETERS

RECORD LEGNTH	=	0.60000 SEC
SAMPLING PERIOD	=	0.10000 MSC
EQUATIONS FORMED	=	120
PRECISION OF DATA	=	3 DECIMAL PLACES
INPUT FUNCTION	=	UNIT STEP

IDENTIFICATION OF UNKNOWN SYSTEM

SYSTEM TRANSFER FUNCTION

POLES	REAL	IMAGINARY	
1	-105.3244028971	0.0	J
2	-1.0000069733	0.0	J
3	-10.0258525504	0.0	J

ZEROES	REAL	IMAGINARY	
1	-3.0005906385	0.0	J
2	-30.4710561855	0.0	J

GAIN CONSTANT = 10.3944911957

SYSTEM STATE VARIABLES (PHASE FORM)

A VECTOR

1	1055.9742969792
2	1171.3179932211
3	116.3502624208

B VECTOR

3	10.3944911957

C VECTOR

1	91.4311659362
2	33.4716468241
3	1.0000000000

PROGRAM PARAMETERS

RECORD LEGNTH	=	1.20000 SEC
SAMPLING PERIOD	=	0.20000 MSC
EQUATIONS FORMED	=	120
PRECISION OF DATA	=	3 DECIMAL PLACES
INPUT FUNCTION	=	UNIT STEP

IDENTIFICATION OF UNKNOWN SYSTEM

SYSTEM TRANSFER FUNCTION

POLES	REAL	IMAGINARY	
1	-107.6394221979	0.0	J
2	-1.0000162038	0.0	J
3	-10.0077785631	0.0	J

ZEROES	REAL	IMAGINARY	
1	-3.0001692900	0.0	J
2	-30.3047449110	0.0	J

GAIN CONSTANT = 10.6635513306

SYSTEM STATE VARIABLES (PHASE FORM)

A VECTOR

1	1077.2489572831
2	1194.8806091150
3	118.6472169649

B VECTOR

3	10.6635513306

C VECTOR

1	90.9193650217
2	33.3049142009
3	1.0000000000

PROGRAM PARAMETERS

RECORD LEGNTH	=	3.00000 SEC
SAMPLING PERIOD	=	0.50000 MSC
EQUATIONS FORMED	=	120
PRECISION OF DATA	=	3 DECIMAL PLACES
INPUT FUNCTION	=	UNIT STEP

EXAMPLE 3.

SYSTEM TO BE IDENTIFIED

SYSTEM TRANSFER FUNCTION

POLES	REAL	IMAGINARY	
1	-1.0000000000	0.0	J
2	-10.0000000000	0.0	J
3	-100.0000000000	0.0	J

ZEROES	REAL	IMAGINARY	
1	-3.0000000000	0.0	J
2	-30.0000000000	0.0	J

GAIN CONSTANT = 10.0000000000

SYSTEM STATE VARIABLES (PHASE FORM)

A VECTOR	
1	1000.0000000000
2	1110.0000000000
3	111.0000000000

B VECTOR	
3	10.0000000000

C VECTOR	
1	90.0000000000
2	33.0000000000
3	1.0000000000

IDENTIFICATION OF UNKNOWN SYSTEM

SYSTEM TRANSFER FUNCTION

POLES	REAL	IMAGINARY	
1	-99.6074714180	0.0	J
2	-1.0627466213	0.0	J
3	-10.0526991957	0.0	J

ZEROES	REAL	IMAGINARY	
1	-3.1109954642	0.0	J
2	-29.8941214051	0.0	J

GAIN CONSTANT = 9.9883327484

SYSTEM STATE VARIABLES (PHASE FORM)

A VECTOR	
1	1064.1536423652
2	1117.8649236164
3	110.7229172350

B VECTOR	
3	9.9883327484

C VECTOR	
1	93.0004760964
2	33.0051168692
3	1.0000000000

PROGRAM PARAMETERS

RECORD LEGNTH	=	0.60000 SEC
SAMPLING PERIOD	=	0.10000 MSC
EQUATIONS FORMED	=	120
PRECISION OF DATA	=	3 DECIMAL PLACES
INPUT FUNCTION	=	GAUSSIAN- SIGMA=0.010 SEC

IDENTIFICATION OF UNKNOWN SYSTEM

SYSTEM TRANSFER FUNCTION

POLES	REAL	IMAGINARY	
1	-98.9033657608	0.0	J
2	-0.9389787239	0.0	J
3	-9.8317762550	0.0	J

ZEROES	REAL	IMAGINARY	
1	-2.8677895731	0.0	J
2	-29.6513565567	0.0	J

GAIN CONSTANT = 9.9508285522

SYSTEM STATE VARIABLES (PHASE FORM)

A VECTOR

1	913.0589327259
2	1074.4957479197
3	109.6741207397

B VECTOR

3	9.9508285522

C VECTOR

1	85.0338511612
2	32.5191461297
3	1.0000000000

PROGRAM PARAMETERS

RECORD LEGNTH	=	0.60000 SEC
SAMPLING PERIOD	=	0.10000 MSC
EQUATIONS FORMED	=	120
PRECISION OF DATA	=	3 DECIMAL PLACES
INPUT FUNCTION	=	GAUSSIAN- SIGMA=0.030 SEC

IDENTIFICATION OF UNKNOWN SYSTEM

SYSTEM TRANSFER FUNCTION

POLES	REAL	IMAGINARY	
1	-99.1859265460	0.0	J
2	-0.7535004508	0.0	J
3	-9.5204582353	0.0	J

ZEROES	REAL	IMAGINARY	
1	-2.5186589038	0.0	J
2	-29.4122032416	0.0	J

GAIN CONSTANT = 9.9813451767

SYSTEM STATE VARIABLES (PHASE FORM)

A VECTOR

1	711.5270631979
2	1026.2057811417
3	109.4598852321

B VECTOR

3	9.9813451767

C VECTOR

1	74.0793075760
2	31.9308621454
3	1.0000000000

PROGRAM PARAMETERS

RECORD LEGNTH	=	0.60000 SEC
SAMPLING PERIOD	=	0.10000 MSC
EQUATIONS FORMED	=	120
PRECISION OF DATA	=	3 DECIMAL PLACES
INPUT FUNCTION	=	GAUSSIAN- SIGMA=0.050 SEC

EXAMPLE 4

SYSTEM TO BE IDENTIFIED

SYSTEM TRANSFER FUNCTION

POLES	REAL	IMAGINARY	
1	-5.0000000000	0.0	J
2	-20.0000000000	0.0	J

ZEROES	REAL	IMAGINARY

GAIN CONSTANT = 300.0000000000

SYSTEM STATE VARIABLES (PHASE FORM)

A VECTOR

1	100.0000000000
2	25.0000000000

B VECTOR

2	300.0000000000

C VECTOR

1	1.0000000000

INITIAL STATE VECTOR

1	1.0000000000
2	0.0

IDENTIFICATION OF UNKNOWN SYSTEM

SYSTEM TRANSFER FUNCTION

POLES	REAL	IMAGINARY	
1	-20.1020603504	0.0	J
2	-0.4280895557	-6.7540893481	J
3	-0.4280895557	6.7540893481	J
4	1.0489471395	0.0	J
5	-4.9974142741	0.0	J

ZEROES	REAL	IMAGINARY	
1	21168.5357639489	0.0	J
2	-0.4681078779	-6.7682935708	J
3	-0.4681078779	6.7682935708	J
4	1.0470216652	0.0	J

GAIN CONSTANT = -0.0141926892

SYSTEM STATE VARIABLES (PHASE FORM)

A VECTOR	
1	-4826.300213
2	3305.020433
3	1059.631159
4	140.5228441
5	24.90670660

B VECTOR	
5	-0.1419268921E-01

C VECTOR	
1	1020181.159
2	-953662.8857
3	2390.647541
4	-21168.64657
5	1.000000000

IDENTIFICATION OF UNKNOWN SYSTEM

INITIAL CONDITION (G) VECTOR

1	0.9983029652
2	25.01148179
3	37.55868385
4	1118.728299
5	-1523.539669

PROGRAM PARAMETERS

RECORD LEGNTH	=	3.00000 SEC
SAMPLING PERIOD	=	0.50000 MSC
EQUATIONS FORMED	=	120
PRECISION OF DATA	=	3 DECIMAL PLACES
INPUT FUNCTION	=	PIECEWISE CONSTANT

IDENTIFICATION OF UNKNOWN SYSTEM

SYSTEM TRANSFER FUNCTION

POLES	REAL	IMAGINARY	
1	-4.9994757762	0.0	J
2	-20.0118644519	0.0	J

ZEROES	REAL	IMAGINARY

GAIN CONSTANT = 300.1457519531

SYSTEM STATE VARIABLES (PHASE FORM)

A VECTOR

1	100.0488316
2	25.01134023

B VECTOR

2	300.1457520

C VECTOR

1	1.000000000

INITIAL CONDITION (G) VECTOR

1	1.001249200
2	25.00380377

EXAMPLE 5

SYSTEM TO BE IDENTIFIED

SYSTEM TRANSFER FUNCTION

POLES	REAL	IMAGINARY
1	-2.0000000000	3.0000000000 J
2	-2.0000000000	-3.0000000000 J
3	-20.0000000000	0.0 J

ZEROES	REAL	IMAGINARY
1	-8.0000000000	0.0 J

GAIN CONSTANT = 160.0000000000

SYSTEM STATE VARIABLES (PHASE FORM)

A VECTOR	
1	260.0000000000
2	93.0000000000
3	24.0000000000

B VECTOR	
3	160.0000000000

C VECTOR	
1	8.0000000000
2	1.0000000000

INITIAL STATE VECTOR	
1	2.0000000000
2	1.0000000000
3	0.0

IDENTIFICATION OF UNKNOWN SYSTEM

SYSTEM TRANSFER FUNCTION

POLES	REAL	IMAGINARY
1	-20.3839954707	0.0 J
2	1.0012938647	-8.8128722093 J
3	1.0012938647	8.8128722093 J
4	-1.9998854338	-3.0017839895 J
5	-1.9998854338	3.0017839895 J

ZEROES	REAL	IMAGINARY
1	-7.6038902797	0.0 J
2	447.1906330529	0.0 J
3	1.2922226696	-9.0382447685 J
4	1.2922226696	9.0382447685 J

GAIN CONSTANT = -0.3623618484

SYSTEM STATE VARIABLES (PHASE FORM)

A VECTOR	
1	20863.16713
2	6906.430945
3	1994.127061
4	124.3802348
5	22.38117861

B VECTOR	
5	-0.3623618484

C VECTOR	
1	-283455.3928
2	-27855.70425
3	-2180.940891
4	-442.1711881
5	1.000000000

IDENTIFICATION OF UNKNOWN SYSTEM

INITIAL CONDITION (G) VECTOR

1	16.99753824
2	388.4269451
3	1684.166654
4	30704.56626
5	95381.34774

PROGRAM PARAMETERS

RECORD LEGNTH	=	3.00000 SEC
SAMPLING PERIOD	=	0.50000 MSC
EQUATIONS FORMED	=	120
PRECISION OF DATA	=	3 DECIMAL PLACES
INPUT FUNCTION	=	PIECEWISE CONSTANT

IDENTIFICATION OF UNKNOWN SYSTEM

SYSTEM TRANSFER FUNCTION

POLES	REAL	IMAGINARY	
1	-20.1316142373	0.0	J
2	-1.9997989237	-3.0011201212	J
3	-1.9997989237	3.0011201212	J

ZEROES	REAL	IMAGINARY	
1	-8.0335677123	0.0	J

GAIN CONSTANT = 160.4525451660

SYSTEM STATE VARIABLES (PHASE FORM)

A VECTOR

1	261.8301183
2	93.52427869
3	24.13121208

B VECTOR

3	160.4525452

C VECTOR

1	8.033567712
2	1.000000000

INITIAL CONDITION (G) VECTOR

1	16.99920347
2	418.2532176
3	1168.376804

EXAMPLE 6

SYSTEM TO BE IDENTIFIED

SYSTEM TRANSFER FUNCTION

POLES	REAL	IMAGINARY	
1	-3.0000000000	0.0	J
2	-45.0000000000	0.0	J

ZEROES	REAL	IMAGINARY	
1	-15.0000000000	0.0	J.

GAIN CONSTANT = 10.0000000000

SYSTEM STATE VARIABLES (PHASE FORM)

A VECTOR

1	135.0000000000
2	48.0000000000

B VECTOR

2	10.0000000000

C VECTOR

1	15.0000000000
2	1.0000000000

INITIAL STATE VECTOR

1	1.0000000000
2	0.0

IDENTIFICATION OF UNKNOWN SYSTEM

SYSTEM TRANSFER FUNCTION

POLES	REAL	IMAGINARY	
1	-44.9825756448	0.0	J
2	0.8248831729	-16.8802782130	J
3	0.8248831729	16.8802782130	J
4	-2.9985948695	0.0	J
5	-4.9097021934	0.0	J

ZEROES	REAL	IMAGINARY	
1	0.8362974237	-16.8739835601	J
2	0.8362974237	16.8739835601	J
3	-15.0390073803	0.0	J
4	-4.9017160580	0.0	J

GAIN CONSTANT = 9.9892787933

SYSTEM STATE VARIABLES (PHASE FORM)

A VECTOR	
1	189152.5939
2	104719.1699
3	15157.98857
4	568.8244217
5	51.24110636

B VECTOR	
5	9.989278793

C VECTOR	
1	21041.07999
2	5568.396359
3	325.7949073
4	18.26812859
5	1.000000000

IDENTIFICATION OF UNKNOWN SYSTEM

INITIAL CONDITION (G) VECTOR

1	14.99958219
2	633.6563568
3	6068.388958
4	183314.0231
5	819995.9356

PROGRAM PARAMETERS

RECORD LEGNTH	=	1.33333 SEC
SAMPLING PERIOD	=	0.22222 MSC
EQUATIONS FORMED	=	120
PRECISION OF DATA	=	4 DECIMAL PLACES
INPUT FUNCTION	=	PIECEWISE CONSTANT

IDENTIFICATION OF UNKNOWN SYSTEM

SYSTEM TRANSFER FUNCTION

POLES	REAL	IMAGINARY	
1	-2.9998001615	0.0	J
2	-44.9468936931	0.0	J

ZEROES	REAL	IMAGINARY	
1	-14.9948012787	0.0	J

GAIN CONSTANT = 9.9912204742

SYSTEM STATE VARIABLES (PHASE FORM)

A VECTOR

1	134.8316990
2	47.94669385

B VECTOR

2	9.991220474

C VECTOR

1	14.99480128
2	1.000000000

INITIAL CONDITION (G) VECTOR

1	14.99876932
2	584.3008098

EXAMPLE 7

SYSTEM TO BE IDENTIFIED

SYSTEM TRANSFER FUNCTION

POLES	REAL	IMAGINARY
1	-1.0000000000	-1.0000000000 J
2	-1.0000000000	1.0000000000 J
3	-50.0000000000	0.0 J

ZEROES	REAL	IMAGINARY
1	-5.0000000000	0.0 J
2	-20.0000000000	0.0 J

GAIN CONSTANT = 5.0000000000

SYSTEM STATE VARIABLES (PHASE FORM)

A VECTOR

1	100.0000000000
2	102.0000000000
3	52.0000000000

B VECTOR

3	5.0000000000

C VECTOR

1	100.0000000000
2	25.0000000000
3	1.0000000000

INITIAL STATE VECTOR

1	2.0000000000
2	1.0000000000
3	0.0

IDENTIFICATION OF UNKNOWN SYSTEM

SYSTEM TRANSFER FUNCTION

POLES	REAL	IMAGINARY	
1	-49.9911920718	0.0	J
2	1.1558667717	-23.4137133914	J
3	1.1558667717	23.4137133914	J
4	-0.9997223909	-0.9974653718	J
5	-0.9997223909	0.9974653718	J

ZEROES	REAL	IMAGINARY	
1	1.2179838023	-23.3753514222	J
2	1.2179838023	23.3753514222	J
3	-19.8075068598	0.0	J
4	-5.1152815779	0.0	J

GAIN CONSTANT = 4.9560489655

SYSTEM STATE VARIABLES (PHASE FORM)

A VECTOR	
1	54789.78240
2	55794.37204
3	28434.85333
4	531.2985138
5	49.67890331

B VECTOR	
5	4.956048965

C VECTOR	
1	55512.80354
2	13408.14537
3	588.5004083
4	22.48682083
5	1.000000000

IDENTIFICATION OF UNKNOWN SYSTEM

INITIAL CONDITION (G) VECTOR

1	225.0001273
2	10975.84397
3	117558.8284
4	6270060.545
5	11259586.57

PROGRAM PARAMETERS

RECORD LEGNTH	=	1.20000 SEC
SAMPLING PERIOD	=	0.20000 MSC
EQUATIONS FORMED	=	120
PRECISION OF DATA	=	4 DECIMAL PLACES
INPUT FUNCTION	=	PIECEWISE CONSTANT

IDENTIFICATION OF UNKNOWN SYSTEM

SYSTEM TRANSFER FUNCTION

POLES	REAL	IMAGINARY	
1	-49.9862042435	0.0	J
2	-0.9995303000	-1.0007669686	J
3	-0.9995303000	1.0007669686	J

ZEROES	REAL	IMAGINARY	
1	-4.9622611725	0.0	J
2	-20.2677062457	0.0	J.

GAIN CONSTANT = 4.9685173035

SYSTEM STATE VARIABLES (PHASE FORM)

A VECTOR

1	100.0021676
2	101.9260468
3	51.98526484

B VECTOR

3	4.968517303

C VECTOR

1	100.5736518
2	25.22996742
3	1.000000000

INITIAL CONDITION (G) VECTOR

1	225.0005829
2	11494.69251
3	20485.03332

EXAMPLE 8

SYSTEM TO BE IDENTIFIED

SYSTEM TRANSFER FUNCTION

POLES	REAL	IMAGINARY	
1	-3.0000000000	0.0	J
2	-8.0000000000	0.0	J
3	-40.0000000000	-5.0000000000	J
4	-40.0000000000	5.0000000000	J

ZEROES	REAL	IMAGINARY	
1	-20.0000000000	0.0	J

GAIN CONSTANT = 160.0000000000

SYSTEM STATE VARIABLES (PHASE FORM)

A VECTOR	
1	39000.0000000000
2	19795.0000000000
3	2529.0000000000
4	91.0000000000

B VECTOR	
4	160.0000000000

C VECTOR	
1	20.0000000000
2	1.0000000000

INITIAL STATE VECTOR	
1	3.0000000000
2	2.0000000000
3	1.0000000000
4	0.0

IDENTIFICATION OF UNKNOWN SYSTEM

SYSTEM TRANSFER FUNCTION

POLES	REAL	IMAGINARY	
1	-40.0576848888	-4.8648939727	J
2	-40.0576848888	4.8648939727	J
3	-8.0002609098	0.0	J
4	-1.4530133153	0.0	J
5	-2.9998483817	0.0	J

ZEROES	REAL	IMAGINARY	
1	-1.4492282436	0.0	J
2	-20.3685768935	0.0	J

GAIN CONSTANT = 157.7509613037

SYSTEM STATE VARIABLES (PHASE FORM)

A VECTOR	
1	56781.06779
2	67897.28681
3	23515.35114
4	2665.954710
5	92.56849238

B VECTOR	
5	157.7509613

C VECTOR	
1	29.51871692
2	21.81780514
3	1.000000000

IDENTIFICATION OF UNKNOWN SYSTEM

INITIAL CONDITION (G) VECTOR

1	62.00108720
2	5780.206688
3	169108.2337
4	1409764.537
5	1708815.026

PROGRAM PARAMETERS

RECORD LEGNTH	=	1.48842 SEC
SAMPLING PERIOD	=	0.24807 MSC
EQUATIONS FORMED	=	120
PRECISION OF DATA	=	5 DECIMAL PLACES
INPUT FUNCTION	=	PIECEWISE CONSTANT

IDENTIFICATION OF UNKNOWN SYSTEM

SYSTEM TRANSFER FUNCTION

POLES	REAL	IMAGINARY
1	-40.0840542053	-4.7409985601 J
2	-40.0840542053	4.7409985601 J
3	-3.0000451598	0.0 J
4	-7.9997100927	0.0 J

ZEROES	REAL	IMAGINARY
1	-19.9584019372	0.0 J

GAIN CONSTANT = 160.7285003662

SYSTEM STATE VARIABLES (PHASE FORM)

A VECTOR

1	39100.17487
2	19844.88825
3	2535.037532
4	91.16786366

B VECTOR

4	160.7285004

C VECTOR

1	19.95840194
2	1.000000000

INITIAL CONDITION (G) VECTOR

1	62.00114277
2	5693.349573
3	160935.4623
4	1176687.962

EXAMPLE 9

SYSTEM TO BE IDENTIFIED

SYSTEM TRANSFER FUNCTION

POLES	REAL	IMAGINARY	
1	-1.0000000000	0.0	J
2	-5.0000000000	0.0	J

ZEROES	REAL	IMAGINARY

GAIN CONSTANT = 10.0000000000

SYSTEM STATE VARIABLES (PHASE FORM)

A VECTOR

1	5.0000000000
2	6.0000000000

B VECTOR

2	10.0000000000

C VECTOR

1	1.0000000000

INITIAL STATE VECTOR

1	1.0000000000
2	0.0

IDENTIFICATION OF UNKNOWN SYSTEM

SYSTEM TRANSFER FUNCTION

POLES	REAL	IMAGINARY	
1	-5.0110029817	0.0	J
2	-2874.4029427054	0.0	J
3	0.0094513173	-2.4186034399	J
4	0.0094513173	2.4186034399	J
5	-0.9999781363	0.0	J

ZEROES	REAL	IMAGINARY	
1	0.0096691675	-2.4198473522	J
2	0.0096691675	2.4198473522	J

GAIN CONSTANT = 28777.0156250000

SYSTEM STATE VARIABLES (PHASE FORM)

A VECTOR	
1	84255.60063
2	100828.6134
3	30926.28200
4	17234.39503
5	2880.395021

B VECTOR	
5	28777.01563

C VECTOR	
1	5.855754701
2	-0.1933833505D-01
3	1.000000000

IDENTIFICATION OF UNKNOWN SYSTEM

INITIAL CONDITION (G) VECTOR

1	1.450317645
2	2876.228032
3	17252.85740
4	16462.65716
5	101232.6568

PROGRAM PARAMETERS

RECORD LEGNTH	=	12.00000 SEC
SAMPLING PERIOD	=	2.00000 MSC
EQUATIONS FORMED	=	120
PRECISION OF DATA	=	5 DECIMAL PLACES
INPUT FUNCTION	=	PIECEWISE CONSTANT

IDENTIFICATION OF UNKNOWN SYSTEM

SYSTEM TRANSFER FUNCTION

POLES	REAL	IMAGINARY	
1	-0.9999961462	0.0	J
2	-5.0000869786	0.0	J

ZEROES	REAL	IMAGINARY

GAIN CONSTANT = 10.0001306534

SYSTEM STATE VARIABLES (PHASE FORM)

A VECTOR

1	5.000067709
2	6.000083125

B VECTOR

2	10.00013065

C VECTOR

1	1.000000000

INITIAL CONDITION (G) VECTOR

1	1.000025784
2	6.000036891

EXAMPLE 10

SYSTEM TO BE IDENTIFIED

SYSTEM TRANSFER FUNCTION

POLES	REAL	IMAGINARY
1	-5.0000000000	-5.0000000000 J
2	-5.0000000000	5.0000000000 J
3	-100.0000000000	0.0 J

ZEROES	REAL	IMAGINARY
1	-20.0000000000	0.0 J

GAIN CONSTANT = 100.0000000000

SYSTEM STATE VARIABLES (PHASE FORM)

A VECTOR

1	5000.0000000000
2	1050.0000000000
3	110.0000000000

B VECTOR

3	100.0000000000

C VECTOR

1	20.0000000000
2	1.0000000000

INITIAL STATE VECTOR

1	2.0000000000
2	1.0000000000
3	0.0

IDENTIFICATION OF UNKNOWN SYSTEM

SYSTEM TRANSFER FUNCTION

POLES	REAL	IMAGINARY	
1	3533.8075084749	0.0	J
2	-25.9614103219	0.0	J
3	-100.0395044818	0.0	J
4	-4.9998977161	-5.0000893933	J
5	-4.9998977161	5.0000893933	J

ZEROES	REAL	IMAGINARY	
1	-19.5630088534	0.0	J
2	-26.9189879242	0.0	J

GAIN CONSTANT =-348534.3750000000

SYSTEM STATE VARIABLES (PHASE FORM)

A VECTOR	
1	-458893162.9
2	-113910225.2
3	-13774844.37
4	-476693.1811
5	-3397.806798

B VECTOR	
5	-348534.3750

C VECTOR	
1	526.6163991
2	46.48199678
3	1.000000000

IDENTIFICATION OF UNKNOWN SYSTEM

INITIAL CONDITION (G) VECTOR

1	40.97649863
2	-139289.1252
3	-19623611.76
4	-535751377.4
5	-3138782043.

PROGRAM PARAMETERS

RECORD LEGNTH	=	0.60000 SEC
SAMPLING PERIOD	=	0.10000 MSC
EQUATIONS FORMED	=	120
PRECISION OF DATA	=	5 DECIMAL PLACES
INPUT FUNCTION	=	PIECEWISE CONSTANT

IDENTIFICATION OF UNKNOWN SYSTEM

SYSTEM TRANSFER FUNCTION

POLES	REAL	IMAGINARY	
1	-99.9959669612	0.0	J
2	-4.9999968528	-5.0000354476	J
3	-4.9999968528	5.0000354476	J

ZEROES	REAL	IMAGINARY	
1	-19.9960841378	0.0	J

GAIN CONSTANT = 100.0136718750

SYSTEM STATE VARIABLES (PHASE FORM)

A VECTOR	
1	4999.830647
2	1049.959363
3	109.9959607

B VECTOR	
3	100.0136719

C VECTOR	
1	19.99608414
2	1.000000000

INITIAL CONDITION (G) VECTOR	
1	40.99999105
2	4529.827485
3	34198.63350

IV. CONCLUSIONS

The method of multiple integrations is a practical and flexible method for identifying lumped parameter, linear, time invariant systems. Complete and accurate identifications can be made on the basis of arbitrary input-output records taken over a short time interval and accurate to only three or four significant figures. The computational requirements of the method are not excessive. The procedure can be implemented by relying entirely on subroutines which are available in most computer center libraries.

At present the technique is limited to comparatively low order systems. When the input-output records are good to three or four significant digits the method will be capable of identifying systems up to about fifth order. This limitation is due primarily to the algorithm used to solve the overdetermined set of linear equations. As better algorithms become available it will be possible to identify higher order systems.

The accuracy of an identification is usually comparable to the number of significant digits in the input-output data. Accuracy depends to a lesser extent on the sampling rate, the nature of the input function driving the system, and the order of the system.

There are several areas where additional research might prove fruitful. Since the input-output records must

be in sampled data form for the computer it might be profitable to reformulate the identification procedure from a sampled system standpoint. Standard techniques could be used to convert the sampled system representation obtained from the identification to a continuous system representation. The ability to identify sampled systems would be a worthwhile extension of this method.

The method of multiple integrations may prove very useful as a tool for approximating high order systems with low order systems. Research could be done to determine the quality of the approximations obtained using this method.

```
C                  DIGITAL COMPUTER PROGRAMS
C
C
C     ..........................................................
C
C     MAIN PROGRAM - LINEAR SYSTEM IDENTIFICATION
C
C     PURPOSE
C        TO IDENTIFY LINEAR TIME INVARIANT SYSTEMS ON
C        THE BASIS OF INPUT-OUTPUT RECORDS
C
C     DESCRIPTION OF PARAMETERS
C        INPUT
C        NP     - ESTIMATED NUMBER OF POLES
C        NZ     - ESTIMATED NUMBER OF ZEROES
C        KPTMAX - NUMBER OF DATA POINTS
C        IPTS   - DATA POINTS INTEGRATED PER LINEAR EQ.
C        T      - TIME
C        R      - INPUT AMPLITUDE AT TIME T
C        C      - OUTPUT AMPLITUDE AT TIME T
C
C     REMARKS
C        (1) OUTPUT WILL CONSIST OF A TRANSFER FUNCTION
C            AND STATE VARIABLE REPRESENTATION OF SYSTEM
C        (2) PROGRAM IS PRESENTLY CONFIGURED TO IDENTIFY
C            SYSTEMS SIMULATED BY SUBROUTINE SYSTEM. IF
C            IT IS DESIRED TO IDENTIFY A PHYSICAL SYSTEM
C            REPLACE SUBROUTINE SYSTEM CALLS WITH APPROP-
C            RIATE READ STATEMENTS.
C
C     SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED
C        WHEN IDENTIFYING PHYSICAL SYSTEMS
C        (1) DLLSQ
C        (2) RTPLSB
C        WHEN IDENTIFYING SIMULATED SYSTEMS
C        (1) DLLSQ
C        (2) RTPLSB
C        (3) SYSTEM
C        (4) EXPAND
C        (5) SUMM
C        (6) DIFF
C        (7) PROD
C        (8) GAUSS3
C        (9) ARRAY
C        (10) MINV
C        (11) ROUND
C
C     METHOD
C        MULTIPLE INTEGRALS OF THE INPUT AND OUTPUT DATA
C        ARE USED TO FORMULATE A SET OF OVERDETERMINED
C        LINEAR EQUATIONS. THESE EQUATIONS ARE THEN SOLVED
C        FOR THE UNKNOWN MODEL PARAMETERS USING THE METHOD
C        OF LEAST SQUARES.
C
C     ..........................................................
C
      REAL*8 TO,TN(11),RO(11),RN(11),CO(11),CN(11),DT2
      REAL*8 A(2600),B(200),X(26),AUX(52),CONV(9)
      REAL*8 AP(10),AZ(10),PRA(9),PIA(9),ZRA(8),ZIA(8)
      INTEGER IPIV(26)
      KPTMAX=5002
      IPTS=50
      NP=4
      NZ=NP-1
    5 CONTINUE
      MEQS=KPTMAX/IPTS
      TO=-1.00
      EPS=10.0**(-35)
      K=1
      M=0
      NP1=NP+1
      NP2=NP+2
```

```
      NZ1=NZ+1
      N=NP+NZ+1
      NI=N+NP
C
C     SET CUMULATIVE INTEGRAL VALUES TO ZERO
C
      DO 10 I=2,NP2
      RO(I)=0.00
   10 CO(I)=0.00
C
C     READ IN INITIAL DATA POINT  (T,R,C)
C
      CALL SYSTEM(TO,RO(1),CO(1))
C
      TOFF=TO
   15 K=K+1
C
C     READ IN NEW DATA POINT   (T,R,C)
C
      CALL SYSTEM(TN(1),RN(1),CN(1))
C
C     UPDATE MULTIPLE INTEGRATIONS
C
      DT2=(TN(1)-TO)*0.5
      DO 20 INT=1,NP1
      RN(INT+1)=(RO(INT)+RN(INT))*DT2+RO(INT+1)
   20 CN(INT+1)=(CO(INT)+CN(INT))*DT2+CO(INT+1)
C
C     FORM A LINEAR EQUATION
C
      IF (K.NE.(K/IPTS)*IPTS) GO TO 35
      M=M+1
      B(M)=CN(2)
      TN(2)=(TN(1)-TOFF)
      DO 25 I=1,NP
      IA=(NP-I)*MEQS+M
      IC=(N+I-1)*MEQS+M
      TN(I+2)=TN(I+1)*TN(2)/FLOAT(I+1)
      A(IA)=-CN(I+2)
   25 A(IC)=TN(I+1)
      DO 30 I=1,NZ1
      IA=(NP+I-1)*MEQS+M
      IRN=NP+3-I
   30 A(IA)=RN(IRN)
C
C     RESET OLD VALUES
C
   35 TO=TN(1)
      DO 40 I=1,NP2
      RO(I)=RN(I)
   40 CO(I)=CN(I)
      IF (M.LT.MEQS) GO TO 15
C
C     SOLVE FOR PARAMETERS BY METHOD OF LEAST SQUARES
C
      CALL DLLSQ(A,B,M,NI,1,X,IPIV,EPS,IER,AUX)
C
C     CALCULATE POLES
C
      AP(1)=1.00
      DO 45 I=1,NP
      J=NP+2-I
   45 AP(J)=X(I)
      CALL RTPLSB(NP,AP,PRA,PIA,CONV,IERPZ)
C
C     CALCULATE ZEROES
C
      GAINI=X(N)
      DO 50 I=NP1,N
   50 X(I)=X(I)/X(N)
      DO 55 I=1,NZ1
      J=N+1-I
```

```
   55 AZ(I)=X(J)
      IF (NZ.EQ.0) GO TO 60
      CALL RTPLSB(NZ,AZ,ZRA,ZIA,CONV,IERPZ)
   60 CONTINUE
C
C     OUTPUT
C
      WRITE(6,914)
      WRITE(6,915) K
      WRITE(6,916) M
      WRITE(6,917) IPTS
      WRITE(6,918) EPS
      WRITE(6,919) AUX(1)
      WRITE(6,920) IER
      WRITE(6,921) (IPIV(I),I=1,NI)
      WRITE(6,900)
      WRITE(6,901)
      WRITE(6,902)
      DO 65 I=1,NP
      WRITE(6,903) I,PRA(I),PIA(I)
   65 CONTINUE
      WRITE(6,904)
      IF (NZ.EQ.0) GO TO 75
      DO 70 I=1,NZ
      WRITE(6,903) I,ZRA(I),ZIA(I)
   70 CONTINUE
   75 CONTINUE
      WRITE(6,905) GAINI
      WRITE(6,906)
      WRITE(6,907)
      DO 80 I=1,NP
      II=NP+2-I
      WRITE(6,910) I,AP(II)
   80 CONTINUE
      WRITE(6,908)
      WRITE(6,910) NP,GAINI
      WRITE(6,909)
      DO 85 I=1,NZ1
      II=NZ+2-I
      WRITE(6,910) I,AZ(II)
   85 CONTINUE
      WRITE(6,913)
      DO 90 I=1,NP
      II=N+I
      WRITE(6,910) I,X(II)
   90 CONTINUE
      GO TO 5
  900 FORMAT(1H1,///,25X,'IDENTIFICATION OF UNKNOWN SYSTEM')
  901 FORMAT(///,12X,'SYSTEM TRANSFER FUNCTION')
  902 FORMAT(//,15X,'POLES',12X,'REAL',13X,'IMAGINARY',/)
  903 FORMAT(17X,I2,7X,G15.8,6X,G15.8,1X,'J',/)
  904 FORMAT(//,15X,'ZEROES',11X,'REAL',13X,'IMAGINARY',/)
  905 FORMAT(//,15X,'GAIN CONSTANT =',G15.8,/)
  906 FORMAT(////,12X,'SYSTEM STATE VARIABLES (PHASE FORM)')
  907 FORMAT(//,15X,'A VECTOR',/)
  908 FORMAT(//,15X,'B VECTOR',/)
  909 FORMAT(//,15X,'C VECTOR',/)
  910 FORMAT(17X,I2,7X,G15.8,/)
  911 FORMAT(1H ,9X,'IER',I5,5X,'EPS',E15.8,5X,'AUX',
     1E16.8,/,9X,11I8)
  912 FORMAT(//,5X,4(E20.9,5X),//,5X,4(E20.9,5X),//)
  913 FORMAT(//,15X,'INITIAL STATE VECTOR',/)
  914 FORMAT(//////,12X,'PROGRAM PARAMETERS',/)
  915 FORMAT(15X,'NUMBER OF DATA POINTS     =',I6,/)
  916 FORMAT(15X,'NUMBER OF EQUATIONS       =',I6,/)
  917 FORMAT(15X,'DATA POINTS PER EQUATION =',I6,/)
  918 FORMAT(15X,'EPS       =',E16.8,/)
  919 FORMAT(15X,'RMS ERROR =',E16.8,/)
  920 FORMAT(15X,'IER       =',I4,/)
  921 FORMAT(15X,'IPIV(I)   =',15(I2,1X),/)
      STOP
      END
```

```
C
C     ...........................................................
C
C     SUBROUTINE SYSTEM
C
C     PURPOSE
C        GENERATES INPUT-OUTPUT RECORDS FOR IDENTIFICA-
C        TION PROGRAM BY SIMULATING A SYSTEM DESCRIBED
C        BY A TRANSFER FUNCTION THAT IS READ IN
C
C     USAGE
C        CALL SYSTEM(T,R,C)
C
C     DESCRIPTION OF PARAMETERS
C        INPUT
C        NP      - NUMBER OF POLES
C        P(I)    - THE VECTOR OF POLES
C        NZ      - NUMBER OF ZEROES
C        Z(I)    - THE VECTOR OF ZEROES
C        GAIN    - THE GAIN CONSTANT
C        OUTPUT
C        T       - TIME
C        R       - INPUT AMPLITUDE AT TIME T
C        C       - OUTPUT AMPLITUDE AT TIME T
C
C     REMARKS
C        (1) INPUT FUNCTION MAY BE CHANGED BY
C            CHANGING ONE CARD IN PROGRAM
C        (2) R AND C ARE ROUNDED TO NA DIGITS
C        (3) ALL FLOATING POINT VARIABLES ARE DECLARED
C            DOUBLE PRECISION (REAL*8).
C
C     SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED
C        (1) SUMM
C        (2) DIFF
C        (3) PROD
C        (4) EXPAND
C        (5) GAUSS3
C        (6) ARRAY
C        (7) MINV
C        (8) ROUND
C
C     METHOD
C        TRANSFER FUNCTION IS CONVERTED TO STATE
C        VARIABLE REPRESENTATION AND INTEGRATED USING
C        TRAPEZOIDAL INTEGRATION
C
C     ...........................................................
C
      SUBROUTINE SYSTEM(T,R,C)
C
      REAL*8 AA(10),CC(10),A(9,9),B(9,1),XXX(9,9)
      REAL*8 PHI(9,9),DEL(9,1),XX(9,1),UU(1,1)
      REAL*8 T,R,C,U,DT,AI(9,9),ZZZ(9,9)
      COMPLEX*16 P(9),Z(8)
C
      IF (T.GE.0.00) GO TO 55
C
C     INPUT POLES, ZEROES, AND GAIN CONSTANT
C
      READ(5,899,END=999) NP
      READ(5,898) (P(I),I=1,NP)
      CALL EXPAND (NP,P,AA)
      DO 5 I=1,NP
      CC(I)=0.00
    5 CONTINUE
      READ(5,899) NZ
      IF (NZ.EQ.0) GO TO 10
      READ(5,898) (Z(I),I=1,NZ)
      CALL EXPAND (NZ,Z,CC)
   10 CONTINUE
      READ(5,898) GAIN
```

```
C
C     SET NA
C
      NA=3
C
C     FIND SHORTEST TIME CONSTANT AND CALCULATE DT
C
      DT=0.00
      DO 15 I=1,NP
      IF (DT.LT.CDABS(P(I))) DT=CDABS(P(I))
   15 CONTINUE
      IF (NZ.EQ.0) GO TO 25
      DO 20 I=1,NZ
      IF (DT.LT.CDABS(Z(I))) DT=CDABS(Z(I))
   20 CONTINUE
   25 DT=1.0/(DT*100.0)
      IF (DT.EQ.0.00) DT=0.00001
C
C     FORM A, B, AND C MATRICES
C
      DO 35 I=1,NP
      DO 30 J=1,NP
      AI(I,J)=0.00
      A(I,J)=0.00
   30 CONTINUE
      B(I,1)=0.00
   35 CONTINUE
      DO 40 I=2,NP
      AI(I,I)=1.00
      A(I-1,I)=DT/2.00
      A(NP,I)=-AA(I)*DT/2.00
   40 CONTINUE
      AI(1,1)=1.00
      A(NP,1)=-AA(1)*DT/2.00
      B(NP,1)=GAIN*DT/2.00
      CC(NZ+1)=1.00
C
C     CALCULATE PHI MATRIX
C
      CALL DIFF(AI,A,ZZZ,NP,NP)
      CALL GAUSS3(NP,EPSS,ZZZ,XXX,KER,9)
      CALL SUMM(AI,A,ZZZ,NP,NP)
      CALL PROD(XXX,ZZZ,PHI,NP,NP,NP)
C
C     CALCULATE DEL MATRIX
C
      CALL PROD(XXX,B,DEL,NP,NP,1)
C
C     DEFINE INITIAL CONDITIONS
C
      K=1
      DO 45 I=1,NP
      XX(I,1)=FLOAT(NP-I)
   45 CONTINUE
      UU(1,1)=0.00
      T=0.00
      R=0.00
      C=0.00
      DO 50 I=1,NP
      C=CC(I)*XX(I,1)+C
   50 CONTINUE
      GO TO 65
C
C     CALCULATE NEW DATA POINT (T,R,C)
C
   55 CONTINUE
      T=DT*FLOAT(K)
C
C     INPUT FUNCTION
C
      U=   +0.1*FLOAT(K/53)+FLOAT(K/403)+FLOAT(K/603)
C
```

```
      UU(1,1)=UU(1,1)+U
      CALL PROD(PHI,XX,XXX,NP,NP,1)
      CALL PROD(DEL,UU,ZZZ,NP,1,1)
      C=0.00
      DO 60 I=1,NP
      XX(I,1)=XXX(I,1)+ZZZ(I,1)
      C=CC(I)*XX(I,1)+C
   60 CONTINUE
      R=U
C
C     ROUND OFF R AND C
C
      CALL ROUND(R,NA)
      CALL ROUND(C,NA)
C
      UU(1,1)=U
      K=K+1
C
      RETURN
C
C     OUTPUT
C
   65 CONTINUE
      WRITE(6,900)
      WRITE(6,901)
      WRITE(6,902)
      DO 70 I=1,NP
      WRITE(6,903) I,P(I)
   70 CONTINUE
      WRITE(6,904)
      IF (NZ.EQ.0) GO TO 80
      DO 75 I=1,NZ
      WRITE(6,903) I,Z(I)
   75 CONTINUE
   80 CONTINUE
      WRITE(6,905) GAIN
      WRITE(6,906)
      WRITE(6,907)
      DO 85 I=1,NP
      WRITE(6,910) I,AA(I)
   85 CONTINUE
      WRITE(6,908)
      WRITE(6,910) NP,GAIN
      WRITE(6,909)
      NZ1=NZ+1
      DO 90 I=1,NZ1
      WRITE(6,910) I,CC(I)
   90 CONTINUE
      WRITE(6,913)
      DO 95 I=1,NP
      WRITE(6,910) I,XX(I,1)
   95 CONTINUE
C
  898 FORMAT(2F10.5)
  899 FORMAT(I1)
  900 FORMAT(1H1,////,25X,'SYSTEM TO BE IDENTIFIED')
  901 FORMAT(///,12X,'SYSTEM TRANSFER FUNCTION')
  902 FORMAT(//,15X,'POLES',12X,'REAL',13X,'IMAGINARY',/)
  903 FORMAT(17X,I2,7X,F14.7,6X,F14.7,1X,'J',/)
  904 FORMAT(//,15X,'ZEROES',11X,'REAL',13X,'IMAGINARY',/)
  905 FORMAT(//,15X,'GAIN CONSTANT =',F14.7,/)
  906 FORMAT(////,12X,'SYSTEM STATE VARIABLES (PHASE FORM)')
  907 FORMAT(//,15X,'A VECTOR',/)
  908 FORMAT(//,15X,'B VECTOR',/)
  909 FORMAT(//,15X,'C VECTOR',/)
  910 FORMAT(17X,I2,7X,F14.7,/)
  913 FORMAT(//,15X,'INITIAL STATE VECTOR',/)
C
      RETURN
  999 CONTINUE
      STOP
      END
```

```
C
C     ..................................................................
C
C        SUBROUTINE DLLSQ
C
C        PURPOSE
C           TO SOLVE LINEAR LEAST SQUARES PROBLEMS, I.E. TO
C           MINIMIZE THE EUCLIDEAN NORM OF B-A*X, WHERE A IS
C           A M BY N MATRIX WITH M NOT LESS THAN N.  IN THE
C           SPECIAL CASE M=N SYSTEMS OF LINEAR EQUATIONS MAY
C           BE SOLVED.
C
C        USAGE
C           CALL DLLSQ(A,B,M,N,L,X,IPIV,EPS,IER,AUX)
C
C        DESCRIPTION OF PARAMETERS
C           A      - DOUBLE PRECISION M BY N MATRIX
C                    (DESTROYED).
C           B      - DOUBLE PRECISION M BY L RIGHT HAND SIDE
C                    MATRIX (DESTROYED).
C           M      - ROW NUMBER OF MATRICES A AND B
C           N      - COLUMN NUMBER OF MATRIX A, ROW NUMBER OF
C                    MATRIX X
C           L      - COLUMN NUMBER OF MATRICES B AND X
C           X      - DOUBLE PRECISION N BY L SOLUTION MATRIX
C           IPIV   - INTEGER OUTPUT VECTOR OF DIMENSION N
C                    WHICH CONTAINS INFORMATION ON COLUMN
C                    INTERCHANGES IN MATRIX A.
C           EPS    - SINGLE PRECISION INPUT PARAMETER WHICH
C                    SPECIFIES A RELATIVE TOLERANCE FOR
C                    DETERMINATION OF RANK OF A.
C           IER    - A RESULTING ERROR PARAMETER
C           AUX    - A DOUBLE PRECISION AUXILIARY STORAGE
C                    ARRAY OF DIMENSION MAX(2*N,L). ON RETURN
C                    FIRST L LOCATIONS OF AUX CONTAIN THE
C                    RESULTING LEAST SQUARES.
C
C        REMARKS
C           (1) NO ACTION BESIDES ERROR MESSAGE IER=-2 IN
C               CASE M LESS THAN N.
C           (2) NO ACTION BESIDES ERROR MESSAGE IER=-1 IN
C               CASE OF A ZERO MATRIX A.
C           (3) IF RANK K OF MATRIX A IS FOUND TO BE LESS
C               THAN N BUT GREATER THAN 0, THE PROCEDURE
C               RETURNS WITH ERROR CODE IER=K INTO CALLING
C               PROGRAM. THE LAST N-K ELEMENTS OF VECTOR IPIV
C               DENOTE THE USELESS COLUMNS IN MATRIX A.
C           (4) IF THE PROCEDURE WAS SUCCESSFUL, ERROR
C               PARAMETER IER IS SET TO ZERO.
C
C        SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED
C           NONE
C
C        METHOD
C           HOUSEHOLDER TRANSFORMATIONS ARE USED TO TRANSFORM
C           MATRIX A TO UPPER TRIANGULAR FORM.  AFTER HAVING
C           APPLIED THE SAME TRANSFORMATIONS TO MATRIX B,  AN
C           APPROXIMATE SOLUTION OF THE PROBLEM IS COMPUTED
C           BY BACK SUBSTITUTION.  FOR REFERANCE, SEE
C           GOLUB, G., NUMERICAL METHODS FOR SOLVING LINEAR
C           LEAST SQUARES PROBLEMS, NUMERISCHE MATHEMATIK,
C           VOL. M, ISS.3 (1965), PP.206-216.
C
C     ..................................................................
C
      SUBROUTINE DLLSQ(A,B,M,N,L,X,IPIV,EPS,IER,AUX)
C
      DIMENSION A(1),B(1),X(1),IPIV(1),AUX(1)
      DOUBLE PRECISION A,B,X,AUX,PIV,H,SIG,BETA,TOL
C
C     ERROR TEST
C
```

```
      IF(M-N)30,1,1
C
C     GENERATICN OF INITIAL VECTOR S(K) (K=1,2,...,N)
C     IN STORAGE LOCATIONS AUX(K) (K=1,2,...,N)
C
    1 PIV=0.D0
      IEND=0
      DO 4 K=1,N
      IPIV(K)=K
      H=0.D0
      IST=IEND+1
      IEND=IEND+M
      DO 2 I=IST,IEND
    2 H=H+A(I)*A(I)
      AUX(K)=H
      IF(H-PIV)4,4,3
    3 PIV=H
      KPIV=K
    4 CONTINUE
C
C     ERROR TEST
C
      IF(PIV)31,31,5
C
C     DEFINE TOLERANCE FOR CHECKING RANK OF A
    5 SIG=DSQRT(PIV)
      TOL=SIG*ABS(EPS)
C
C
C     DECOMPOSITICN LOOP
      LM=L*M
      IST=-M
      DO 21 K=1,N
      IST=IST+M+1
      IEND=IST+M-K
      I=KPIV-K
      IF(I)8,8,6
C
C     INTERCHANGE K-TH COLUMN OF A WITH KPIV-TH IN CASE
C     KPIV.GT.K.
    6 H=AUX(K)
      AUX(K)=AUX(KPIV)
      AUX(KPIV)=H
      ID=I*M
      DO 7 I=IST,IEND
      J=I+ID
      H=A(I)
      A(I)=A(J)
    7 A(J)=H
C
C     CCMPUTATICN OF PARAMETER SIG
    8 IF(K-1)11,11,9
    9 SIG=0.D0
      DO 10 I=IST,IEND
   10 SIG=SIG+A(I)*A(I)
      SIG=DSQRT(SIG)
C
C     TEST ON SINGULARITY
      IF(SIG-TOL)32,32,11
C
C     GENERATE CORRECT SIGN OF PARAMETER SIG
   11 H=A(IST)
      IF(H)12,13,13
   12 SIG=-SIG
C
C     SAVE INTERCHANGE INFORMATION
   13 IPIV(KPIV)=IPIV(K)
      IPIV(K)=KPIV
C
C     GENERATION OF VECTOR UK IN K-TH COLUMN OF MATRIX
C     A AND OF PARAMETER BETA
      BETA=H+SIG
```

```
      A(IST)=BETA
      BETA=1.D0/(SIG*BETA)
      J=N+K
      AUX(J)=-SIG
      IF(K-N)14,19,19
C
C     TRANSFORMATION OF MATRIX A
   14 PIV=0.D0
      ID=0
      JST=K+1
      KPIV=JST
      DO 18 J=JST,N
      ID=ID+M
      H=0.D0
      DO 15 I=IST,IEND
      II=I+ID
   15 H=H+A(I)*A(II)
      H=BETA*H
      DO 16 I=IST,IEND
      II=I+ID
   16 A(II)=A(II)-A(I)*H
C
C     UPDATING OF ELEMENT S(J) STORED IN LOCATION AUX(J)
      II=IST+ID
      H=AUX(J)-A(II)*A(II)
      AUX(J)=H
      IF(H-PIV)18,18,17
   17 PIV=H
      KPIV=J
   18 CONTINUE
C
C     TRANSFORMATION OF RIGHT HAND SIDE MATRIX B
   19 DO 21 J=K,LM,M
      H=0.D0
      IEND=J+M-K
      II=IST
      DO 20 I=J,IEND
      H=H+A(II)*B(I)
   20 II=II+1
      H=BETA*H
      II=IST
      DO 21 I=J,IEND
      B(I)=B(I)-A(II)*H
   21 II=II+1
C     END OF DECOMPOSITION LOOP
C
C
C     BACK SUBSTITUTION AND BACK INTERCHANGE
      IER=0
      I=N
      LN=L*N
      PIV=1.D0/AUX(2*N)
      DO 22 K=N,LN,N
      X(K)=PIV*B(I)
   22 I=I+M
      IF(N-1)26,26,23
   23 JST=(N-1)*M+N
      DO 25 J=2,N
      JST=JST-M-1
      K=N+N+1-J
      PIV=1.D0/AUX(K)
      KST=K-N
      ID=IPIV(KST)-KST
      IST=2-J
      DO 25 K=1,L
      H=B(KST)
      IST=IST+N
      IEND=IST+J-2
      II=JST
      DO 24 I=IST,IEND
      II=II+M
   24 H=H-A(II)*X(I)
```

```
      I=IST-1
      II=I+ID
      X(I)=X(II)
      X(II)=PIV*H
   25 KST=KST+M
C
C
C     COMPUTATION OF LEAST SQUARES
   26 IST=N+1
      IEND=0
      DO 29 J=1,L
      IEND=IEND+M
      H=0.D0
      IF(M-N)29,29,27
   27 DO 28 I=IST,IEND
   28 H=H+B(I)*B(I)
      IST=IST+M
   29 AUX(J)=H
      RETURN
C
C     ERROR RETURN IN CASE M LESS THAN N
   30 IER=-2
      RETURN
C
C     ERROR RETURN IN CASE OF ZERO-MATRIX A
   31 IER=-1
      RETURN
C
C     ERROR RETURN IN CASE OF RANK OF MATRIX A LESS THAN N
   32 IER=K-1
      RETURN
      END
C
C     ..................................................................
C
C     SUBROUTINE ROUND
C
C     PURPOSE
C        TO ROUND OFF A NUMBER TO A SPECIFIED NUMBER OF
C        SIGNIFICANT DIGITS
C
C     USAGE
C        CALL ROUND(A,N)
C
C     DESCRIPTION OF PARAMETERS
C        A       - NUMBER TO BE ROUNDED
C        N       - SIGNIFICANT DIGITS TO BE RETAINED
C
C     REMARKS
C        NONE
C
C     SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED
C        NONE
C
C     ..................................................................
C
      SUBROUTINE ROUND(A,N)
      REAL*8 A
      X=A
      IF (X.EQ.0.0) GO TO 1
      SIGN=+1.0
      IF (X.LT.0.0) SIGN=-1.0
      L=ALOG10(ABS(X))+1.00
      Y=X*10.0**(N-L)
      I=Y
      Z=I
      IF ((Y-Z).GT.0.5) I=I+1
      Z=I
      A=SIGN*Z*10.0**(L-N)
    1 CONTINUE
      RETURN
      END
```

```
C
C     ..................................................................
C
C        SUBROUTINE RTPLSB
C
C        PURPOSE
C           TO FIND THE ROOTS, BOTH REAL AND COMPLEX, OF A
C           POLYNOMIAL WITH REAL COEFFICIENTS USING BOTH
C           BAIRSTOW AND NEWTON-RAPHSON METHODS.
C
C        USAGE
C           CALL RTPLSB(N,A,U,V,CONV,IER)
C
C        DESCRIPTION OF PARAMETERS
C           INPUT
C           N      - DEGREE OF POLYNOMIAL
C           A      - COEFFICIENT VECTOR OF POLYNOMIAL
C           OUTPUT
C           U      - VECTOR OF REAL PARTS OF ROOTS
C           V      - VECTOR OF IMAGINARY PARTS OF ROOTS
C           CONV   - CONVERGENCE INDICATORS FOR EACH ROOT
C           IER    - ERROR INDICATOR
C                    =0 , N IS WITHIN BOUNDS
C                    =1 , N IS LESS THAN ONE
C
C        REMARKS
C           (1) FOR PROBLEMS WITH NONMULTIPLE ROOTS, THE
C               APPROXIMATE NUMBER OF SIGNIFICANT DIGITS OF
C               EACH PART OF EACH ROOT WILL APPEAR AS THE
C               EXPONENT OF THE CORRESPONDING ENTRY IN CONV
C           (2) ACCURACY MAY BE LESS THAN INDICATED BY CONV
C               IF THERE ARE MULTIPLE ROOTS
C
C        SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED
C           NONE
C
C     ..................................................................
C
      SUBROUTINE RTPLSB (N,A,U,V,CONV,IER)
      IMPLICIT REAL*8 (A-H), REAL*8 (O-Z)
      DIMENSION A(1),U(1),V(1),CONV(1),B(53)
      DIMENSION C(53),D(53),E(53),H(53)
      KF=10
      L=25
      IER=0
      NN=N
      K=0
      N1=N+1
      IZF=0
      IF(NN) 51,51,52
   52 DO 11 I=1,N1
      IF(A(I)) 15,12,15
   15 CONTINUE
      IZF=1
      GO TO 11
   12 CONTINUE
      IF (IZF-1) 13,14,13
   13 NN=NN-1
      K=K+1
   11 CONTINUE
   14 NP3=NN+3
      N3=NP3
      DO 10 I=1,N3
      CONV(I)=0.
      U(I)=0.
   10 V(I)=0.
      B(2)=0.0
      B(1)=0.0
      C(2)=0.0
      C(1)=0.0
      D(2)=0.0
      E(2)=0.0
```

```
      H(2)=0.0
      DO 101 J=3,NP3
      H(J)=A(J+K-2)
101   CONTINUE
      T=1.0
      SK=10.0**KF
150   IF(H(NP3)) 200,151,200
151   U(NP3)=0.0
      V(NP3)=0.
      CONV(NP3)=SK
      NP3=NP3-1
      IF(NP3) 51,51,150
200   IF(NP3-3) 54,54,201
201   PS=0.0
      QS=0.0
      PT=0.0
      QT=0.0
      S=0.0
      REV=1.0
      SK=10.0**KF
      IF(NP3-4) 51,202,203
202   R=-H(4)/H(3)
      GO TO 500
203   DO 207 J=3,NP3
      IF(H(J))204,207,204
204   S=S+DLOG(DABS(H(J)))
207   CONTINUE
      FPN1=N+1
      S=DEXP(S/FPN1)
      DO 208 J=3,NP3
      H(J)=H(J)/S
208   CONTINUE
      IF(DABS(H(4)/H(3))-DABS(H(NP3-1)/H(NP3)))250,252,252
250   CONTINUE
      T=-T
      M=(NP3-4)/2 + 3
      DO 251 J=3,M
      S=H(J)
      JJ=NP3-J+3
      H(J)=H(JJ)
251   H(JJ)=S
252   IF(QS) 253,254,253
253   CONTINUE
      P=PS
      Q=QS
      GO TO 300
254   HH2=H(NP3-2)
      IF(HH2) 256,255,256
255   Q=1.0
      P=-2.0
      GO TO 257
256   Q=H(NP3)/HH2
      P=(H(NP3-1)-Q*H(NP3-3))/HH2
257   IF(NP3-5)258,550,258
258   R=0.0
300   CONTINUE
      DO 490 I=1,L
      DO 351 J=3,NP3
      B(J)=H(J)-P*B(J-1)-Q*B(J-2)
      C(J)=B(J)-P*C(J-1)-Q*C(J-2)
351   CONTINUE
      IF(H(NP3-1))352,400,352
352   CONTINUE
      IF (B(NP3-1)) 353,400,353
353   AVHB1=DABS(H(NP3-1)/B(NP3-1))
      IF(AVHB1-SK)450,354,354
354   B(NP3)=H(NP3)-Q*B(NP3-2)
400   IF(B(NP3))401,550,401
401   AVHB2=DABS(H(NP3)/B(NP3))
      IF(SK-AVHB2)550,450,450
450   DO 451 J=3,NP3
      D(J)=H(J)+R*D(J-1)
```

```
      E(J)=D(J)+R*E(J-1)
  451 CONTINUE
      IF(D(NP3))452,500,452
  452 CONTINUE
      AVHD3=DABS(H(NP3)/D(NP3))
      IF(SK-AVHD3)500,453,453
  453 CC2=C(NP3-2)
      CC3=C(NP3-3)
      C(NP3-1)=-P*CC2-Q*CC3
      CC1=C(NP3-1)
      S=CC2*CC2-CC1*CC3
      IF(S)455,454,455
  454 CONTINUE
      P=P-2.00
      Q=Q*(Q+1.0)
      GO TO 456
  455 P=P+(B(NP3-1)*CC2-B(NP3)*CC3)/S
      Q=Q+(-B(NP3-1)*CC1+B(NP3)*CC2)/S
  456 IF(E(NP3-1))458,457,458
  457 R=R-1.0
      GO TO 490
  458 R=R-D(NP3)/E(NP3-1)
  490 CONTINUE
      PS=PT
      QS=QT
      PT=P
      QT=Q
      IF(REV)491,492,492
  491 SK=SK/10.0
  492 REV=-REV
      GO TO 250
  500 IF(T)501,502,502
  501 R=1.0/R
  502 NP=NP3-3
      U(NP)=R
      V(NP)=0.0
      CONV(NP)=SK
      NP3=NP3-1
      DO 503 J=3,NP3
  503 H(J)=D(J)
      IF(NP3-3)300,51,300
  550 IF(T)551,552,552
  551 P=P/Q
      Q=1.0/Q
  552 PP2=P/2.0
      QMPSQ=Q-PP2*PP2
      IF(QMPSQ)554,554,553
  553 NP=NP3-3
      U(NP)=-PP2
      U(NP-1)=-PP2
      S=DSQRT(QMPSQ)
      V(NP)=S
      V(NP-1)=-S
      GO TO 561
  554 S=DSQRT(-QMPSQ)
      NP=NP3-3
      IF(P)555,556,556
  555 U(NP)=-PP2+S
      GO TO 557
  556 U(NP)=-PP2-S
  557 U(NP-1)=Q/U(NP)
      V(NP)=0.0
      V(NP-1)=0.0
  561 CONV(NP)=SK
      CONV(NP-1)=SK
      NP3=NP3-2
      DO 558 J=3,NP3
  558 H(J)=B(J)
      GO TO 200
   51 IER=1
   54 RETURN
      END
```

```
C
C     ..............................................................
C
C        SUBROUTINE ARRAY
C
C        PURPOSE
C           CONVERT DATA ARRAY FROM SINGLE TO DOUBLE DIMEN-
C           SION OR VICE VERSA.  THIS SUBROUTINE IS USED TO
C           LINK THE USER PROGRAM WHICH HAS DOUBLE DIMENSION
C           ARRAYS AND THE SSP SUBROUTINES WHICH OPERATE
C           ON ARRAYS OF DATA IN A VECTOR FASHION.
C
C        USAGE
C           CALL ARRAY (MODE,I,J,N,M,S,D)
C
C        DESCRIPTION OF PARAMETERS
C           MODE    - CODE INDICATING TYPE OF CONVERSION
C                     =1 - FROM SINGLE TO DUUBLE PRECISION
C                     =2 - FROM DOUBLE TO SINGLE PRECISION
C           I       - NUMBER OF ROWS IN ACTUAL DATA MATRIX
C           J       - NUMBER OF COLUMNS IN ACTUAL DATA MATRIX
C           N       - NUMBER OF ROWS SPECIFIED FOR THE MATRIX
C                     D IN DIMENSION STATEMENT
C           S       - IF MODE=1, THIS VECTOR IS INPUT WHICH
C                     CONTAINS THE ELEMENTS OF A DATA MATRIX
C                     OF SIZE I BY J. COLUMN I+1 OF DATA
C                     MATRIX FOLLOWS COLUMN I, ETC. IF MODE=2
C                     THIS VECTOR IS OUTPUT REPRESENTING A
C                     DATA MATRIX OF SIZE I BY J CONTAINING
C                     ITS COLUMNS CONSECUTIVELY.  THE LEGNTH
C                     OF S IS IJ=I*J.
C           D       - IF MODE=1, THIS MATRIX OF SIZE N BY M
C                     IS OUTPUT, CONTAINING A DATA MATRIX OF
C                     SIZE I BY J IN THE FIRST I ROWS AND J
C                     COLUMNS. IF MODE=2, THIS N BY M MATRIX
C                     IS INPUT CONTAINING A DATA MATRIX OF
C                     SIZE I BY J IN THE FIRST I ROWS AND J
C                     COLUMNS
C
C        REMARKS
C           VECTOR S CAN BE IN THE SAME LOCATION AS MATRIX
C           D.  VECTOR S IS REFERANCED AS A MATRIX IN OTHER
C           SSP ROUTINES, SINCE IT CONTAINS A DATA MATRIX.
C           THIS ROUTINE CONVERTS ONLY GENERAL DATA MATRICES
C           ( STORAGE MODE 0 )
C
C        SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED
C           NONE
C
C     ..............................................................
C
      SUBROUTINE ARRAY(MODE,I,J,N,M,S,D)
C
      DIMENSION S(1),D(1)
      REAL*8 S,D
C
      NI=N-I
C
C        TEST TYPE OF CONVERSION
C
      IF(MODE-1) 100,100,120
C
C        CONVERT FROM SINGLE TO DOUBLE DIMENSION
C
  100 IJ=I*J+1
      NM=N*J+1
      DO 110 K=1,J
      NM=NM-NI
      DO 110 L=1,I
      IJ=IJ-1
      NM=NM-1
  110 D(NM)=S(IJ)
```

```
      GO TO 140
C
C     CONVERT FROM DOUBLE TO SINGLE DIMENSION
C
  120 IJ=0
      NM=0
      DO 130 K=1,J
      DO 125 L=1,I
      IJ=IJ+1
      NM=NM+1
  125 S(IJ)=D(NM)
  130 NM=NM+NI
  140 RETURN
C
      END
C
C     ..................................................................
C
C     SUBROUTINE GAUSS3
C
C     PURPOSE
C        INVERT A DOUBLE PRECISION MATRIX BY THE GAUSS-
C        JORDAN METHOD.  THIS ROUTINE IS A DOUBLE PRECIS-
C        ION VERSION OF SSP ROUTINE MINV USING F1-NPGS-
C        GAUSS3 (F-63) CALLING SEQUENCE
C
C     USAGE
C
C     DESCRIPTION OF PARAMETERS
C        N       - ORDER OF MATRIX
C        EPS     - DUMMY PARAMETER NOT USED BY GAUSS3
C        A       - TWO DIMENSIONAL ARRAY CONTAINING MATRIX
C                  TO BE INVERTED
C        X       - TWO DIMENSIONAL ARRAY CONTAINING INVERTED
C                  MATRIX
C        KER     - ERROR FLAG
C                  =1 INDICATES NO ERRORS
C                  =2 INDICATES MATRIX IS SINGULAR
C        K       - ROW AND COLUMN DIMENSION OF A AND X IN
C                  USERS PROGRAM
C
C     SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED
C        (1) MINV
C        (2) ARRAY
C
C     REMARKS
C        ALL FLOATING POINT VARIABLES ARE DOUBLE PRECISION
C        (REAL*8).  IF N IS GREATER THAN 50, THE DIMENSION
C        OF ARRAYS L,M, AND Y MUST BE CHANGED TO BE
C        GREATER THAN OR EQUAL TO N.
C
C     ..................................................................
C
      SUBROUTINE GAUSS3(N,EPS,A,X,KER,K)
      REAL*8 A,X,Y,D
      DIMENSION A(1),X(1),L(50),M(50),Y(50,50)
      DO 1 I=1,N
      DO 1 J=1,N
      IND=(I-1)*K+J
    1 Y(I,J)=A(IND)
      KER=1
      N2=2*N
      CALL ARRAY(2,N,N,50,50,Y,Y)
      CALL MINV(Y,N,D,L,M)
      CALL ARRAY(1,N,N,50,50,Y,Y)
      IF(D.EQ.0.) KER=2
      DO 2 I=1,N
      DO 2 J=1,N
      IND=(I-1)*K+J
    2 X(IND)=Y(I,J)
      RETURN
      END
```

```
C
C     ..................................................................
C
C        SUBROUTINE MINV
C
C        PURPOSE
C           INVERT A DOUBLE PRECISION MATRIX BY THE GAUSS-
C           JORDAN METHOD.
C
C        USAGE
C           CALL MINV(A,N,D,L,M)
C
C        DESCRIPTION OF PARAMETERS
C           A      - INPUT MATRIX, DESTROYED IN COMPUTATION
C                    AND REPLACED BY RESULTANT INVERSE.
C           N      - ORDER OF MATRIX A
C           D      - RESULTANT DETERMINANT
C           L      - WORK VECTOR OF LEGNTH N
C           M      - WORK VECTOR OF LEGNTH N
C
C        SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED
C           NONE
C
C        REMARKS
C           NONE
C
C     ..................................................................
C
      SUBROUTINE MINV(A,N,D,L,M)
      DIMENSION A(1),L(1),M(1)
      DOUBLE PRECISION A,D,BIGA,HOLD
C
C        SEARCH FOR LARGEST ELEMENT
C
      D=1.0
      NK=-N
      DO 80 K=1,N
      NK=NK+N
      L(K)=K
      M(K)=K
      KK=NK+K
      BIGA=A(KK)
      DO 20 J=K,N
      IZ=N*(J-1)
      DO 20 I=K,N
      IJ=IZ+I
   10 IF(DABS(BIGA)-DABS(A(IJ))) 15,20,20
   15 BIGA=A(IJ)
      L(K)=I
      M(K)=J
   20 CONTINUE
C
C        INTERCHANGE ROWS
C
      J=L(K)
      IF(J-K) 35,35,25
   25 KI=K-N
      DO 30 I=1,N
      KI=KI+N
      HOLD=-A(KI)
      JI=KI-K+J
      A(KI)=A(JI)
   30 A(JI) =HOLD
C
C        INTERCHANGE COLUMNS
C
   35 I=M(K)
      IF(I-K) 45,45,38
   38 JP=N*(I-1)
      DO 40 J=1,N
      JK=NK+J
      JI=JP+J
```

```
      HOLD=-A(JK)
      A(JK)=A(JI)
   40 A(JI) =HOLD
C
C        DIVIDE COLUMN BY MINUS PIVOT (VALUE OF PIVOT
C        ELEMENT IS CONTAINED IN BIGA)
C
   45 IF(BIGA) 48,46,48
   46 D=0.0
      RETURN
   48 DO 55 I=1,N
      IF(I-K) 50,55,50
   50 IK=NK+I
      A(IK)=A(IK)/(-BIGA)
   55 CONTINUE
C
C        REDUCE MATRIX
C
      DO 65 I=1,N
      IK=NK+I
      HOLD=A(IK)
      IJ=I-N
      DO 65 J=1,N
      IJ=IJ+N
      IF(I-K) 60,65,60
   60 IF(J-K) 62,65,62
   62 KJ=IJ-I+K
      A(IJ)=HOLD*A(KJ)+A(IJ)
   65 CONTINUE
C
C        DIVIDE ROW BY PIVOT
C
      KJ=K-N
      DO 75 J=1,N
      KJ=KJ+N
      IF(J-K) 70,75,70
   70 A(KJ)=A(KJ)/BIGA
   75 CONTINUE
      D=D*BIGA
C
C        REPLACE PIVOT BY RECIPROCAL
C
      A(KK)=1.0/BIGA
   80 CONTINUE
C
C        FINAL ROW AND COLUMN INTERCHANGE
C
      K=N
  100 K=(K-1)
      IF(K) 150,150,105
  105 I=L(K)
      IF(I-K) 120,120,108
  108 JQ=N*(K-1)
      JR=N*(I-1)
      DO 110 J=1,N
      JK=JQ+J
      HOLD=A(JK)
      JI=JR+J
      A(JK)=-A(JI)
  110 A(JI) =HOLD
  120 J=M(K)
      IF(J-K) 100,100,125
  125 KI=K-N
      DO 130 I=1,N
      KI=KI+N
      HOLD=A(KI)
      JI=KI-K+J
      A(KI)=-A(JI)
  130 A(JI) =HOLD
      GO TO 100
  150 RETURN
      END
```

```
C
C     ..................................................................
C
C        SUBROUTINE PROD
C
C        PURPOSE
C           TO COMPUTE THE PRODUCT OF TWO MATRICES
C
C        USAGE
C           CALL PROD(A,B,C,N,M,L)
C
C        DESCRIPTION OF PARAMETERS
C           A       - FIRST INPUT MATRIX
C           B       - SECOND INPUT MATRIX
C           C       - PRODUCT OF A AND B
C           N       - NUMBER OF ROWS IN A AND C
C           M       - NUMBER OF COLUMNS IN A AND ROWS IN B
C           L       - NUMBER OF COLUMNS IN B AND C
C
C        REMARKS
C           NONE
C
C        SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED
C           NONE
C
C     ..................................................................
C
      SUBROUTINE PROD(A,B,C,N,M,L)
      REAL*8 A(9,9),B(9,9),C(9,9)
      DO 1 I=1,N
      DO 1 J=1,L
      C(I,J)=0.00
      DO 1 K=1,M
    1 C(I,J)=C(I,J)+A(I,K)*B(K,J)
      RETURN
      END
C
C
C
C     ..................................................................
C
C        SUBROUTINE SUMM
C
C        PURPOSE
C           ADD TWO MATRICES
C
C        USAGE
C           CALL SUMM(A,B,C,M,N)
C
C        DESCRIPTION OF PARAMETERS
C           A       - NAME OF FIRST INPUT MATRIX
C           B       - NAME OF SECOND INPUT MATRIX
C           C       - NAME OF OUTPUT MATRIX
C           M       - NUMBER OF ROWS IN A,B, AND C
C           N       - NUMBER OF COLUMNS IN A,B, AND C
C
C        REMARKS
C           NONE
C
C        SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED
C           NONE
C
C     ..................................................................
C
      SUBROUTINE SUMM(A,B,C,M,N)
      REAL*8 A(9,9),B(9,9),C(9,9)
      DO 1 I=1,M
      DO 1 J=1,N
    1 C(I,J)=A(I,J)+B(I,J)
      RETURN
      END
C
```

```
C
C     ..................................................................
C
C        SUBROUTINE DIFF
C
C        PURPOSE
C           SUBTRACT ONE MATRIX FROM ANOTHER
C
C        USAGE
C           CALL DIFF(A,B,C,M,N)
C
C        DESCRIPTICN OF PARAMETERS
C           A       - FIRST INPUT MATRIX
C           B       - SECOND INPUT MATRIX
C           C       - OUTPUT MATRIX EQUALS A - B
C           M       - NUMBER OF ROWS IN A,B, AND C
C           N       - NUMBER OF COLUMNS IN A,B, AND C
C
C        REMARKS
C           NONE
C
C        SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED
C           NONE
C
C     ..................................................................
C
      SUBROUTINE DIFF(A,B,C,M,N)
      REAL*8 A(9,9),B(9,9),C(9,9)
      DO 1 I=1,M
      DO 1 J=1,N
    1 C(I,J)=A(I,J)-B(I,J)
      RETURN
      END
C
C
C
C     ..................................................................
C
C        SUBROUTINE EXPAND
C
C        PURPOSE
C           TO COMPUTE THE REAL COEFFICIENTS OF AN N-TH
C           DEGREE PCLYNOMIAL GIVEN N COMPLEX ROOTS
C
C        USAGE
C           CALL EXPAND(N,R,A)
C
C        DESCRIPTION OF PARAMETERS
C           N       - DEGREE OF POLYNOMIAL
C           R       - VECTOR CF COMPLEX ROOTS
C           A       - COEFFICIENT VECTOR
C
C        REMARKS
C           NONE
C
C        SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED
C           NONE
C
C     ..................................................................
C
      SUBROUTINE EXPAND(N,R,A)
      REAL*8 A(10)
      CCMPLEX*16 R(9),Q(9)
      IF (N-1) 1,2,3
    1 A(1)=1.00
      RETURN
    2 A(1)=-REAL(R(1))
      A(2)=1.00
      RETURN
    3 Q(1)=-R(1)
      Q(2)=1.00
      DO 5 J=2,N
```

```
      Q(J+1)=1.00
      JJ=J-1
      DO 4 I=1,JJ
      K=J-I
    4 Q(K+1)=Q(K)-Q(K+1)*R(J)
    5 Q(1)=-Q(1)*R(J)
    6 N1=N+1
      DO 7 L=1,N1
    7 A(L)=REAL(Q(L))
      RETURN
      END
```

BIBLIOGRAPHY

1. Nieman, R. E., Fisher, D. G., and Seborg, D. E., "A Review of Process Identification and Parameter Estimation Techniques," International Journal of Control, V. 13, No. 2, p. 209-264, May 1971.

2. Eykhoff, P., van der Grinten, M., Kwakernaai, H., and Veltman, B., "Survey-Systems Modeling and Identification," Third Congress of I.F.A.C., London, June 1966.

3. Mishkin, E., and Haddad, R. A., "Identification and Command Problems in Adaptive Systems," IRE Trans. on Automatic Control, V. AC-4, No. 2, p. 121-131, November 1959.

4. Levin, M. J., "Optimum Estimation of Impulse Response in the Presence of Noise," IRE Trans. on Circuit Theory, V. CT-7, No. 1, p. 50-56, March 1960.

5. Kerr, R. B., and Surber, Jr., N. H., "Precision of Impulse Response Identification Based on Short, Normal Operating Records," IRE Trans. on Automatic Control, V. AC-6, No. 2, p. 173-182, May 1961.

6. Turin, G. L., "On the Estimation in the Presence of Noise of the Impulse Response of a Random, Linear Filter," IRE Trans. on Information Theory, V. IT-3, No. 1, p. 5-10, March 1957.

7. Lichtenberger, W. W., "On the Identification of Linear Processes," IRE Trans. on Circuit Theory, V. CT-7, No. 1, p. 78-79, March 1960.

8. Lichtenberger, W. W., "A Technique of Linear System Identification Using Correlating Filters," IRE Trans. on Automatic Control, V. AC-9, No. 2, p. 183-199, May 1961.

9. Hill, J. D., and McMurtry, "An Application of Digital Computers to Linear System Identification," IEEE Trans. on Automatic Control, V. AC-9, No. 5, p. 536-538, October 1964.

10. Lindenlaub, J. C., and Cooper, G. R., "Noise Limitations of System Identification Techniques," IEEE Trans. on Automatic Control, V. AC-10, No. 1, p. 43-48, January 1963.

11. Kumar, K. S. P., and Sridhar, R., "On the Identification of Control Systems by the Quasi-Linearization Method," IEEE Trans. on Automatic Control, V. AC-9, No. 2, p. 151-154, April 1964.

12. Nagumo, J., and Noda, A., "A Learning Method for System Identification," IEEE Trans. on Automatic Control, V. AC-12, No. 3, p. 282-287, June 1967.

13. Bass, R., Testing Function Techniques in System Identification, Ph.D. Dissertation, University of California, Berkeley, March 1969.

14. Astrom, K. J., and Bohlin, T., "Numerical Identification of Linear Dynamic Systems from Normal Operating Records," IFAC Symposium - Theory of Self-Adaptive Systems, Teddington, 1965.

15. Peterka, V., "New Approach to the Identification of Discrete Dynamic Systems," Kybernetika, V. 4, No. 5, 1968.

16. Peterka, V., and Smuk, K., "On-Line Estimation of Dynamic Model Parameters from Input-Output Data," IFAC Congress, Warsaw, 1969.

17. Ho, B. L., and Kalman, R. E., "Effective Construction of Linear State Variable Models from Input/Output Functions," Sonderdruck aus Regelungstechnik, V. 14, No. 12, p. 545-548, 1966.

18. Eldem, I., Identification of Plant Dynamics Using Ho's Algorithm, MS Thesis, United States Naval Postgraduate School, Monterey, California, 1969.

19. Shinbrot, M., "On the Analysis of Linear and Nonlinear Systems," Trans. of the ASME, V. 79, No. 3, p. 547-552, April 1957.

20. Mowery, V. O., "Least Squares Recursive Differential-Correlation Estimation in Nonlinear Problems," IEEE Trans. on Automatic Control, V. AC-10, No. 4, p. 399-407, October 1965.

21. Fairman, F. W., Process Parameter Estimation by the Moment Function Method, Ph.D. Dissertation, University of Pennsylvania, 1968.

22. Bellman, R. E., Kagiwada, H., and Kalaba, R. E., "Quasi-Linearization, System Identification and Prediction," Rand Corp. Report RM-3821-PR, August 1963.

23. Hsieh, H. C., "Adaptive Control Systems," Advances in Control Systems, V. 2, Academic Press Inc., New York, New York, 1965.

24. Simpson, H. R., "A Sampled Data Nonlinear Filer," Proc. of the IEEE, V. 112, No. 6, p. 1187-1196, June 1965.

25. Bose, A. G., "A Theory for the Experimental Determination of Optimum Nonlinear Systems," IRE National Convention Record, Part 4, p. 21-30, 1956.

26. Bose, A. G., "Nonlinear System Characterization and Optimization," IRE Trans. on Circuit Theory, Special Supplement, p. 30-40, 1959.

27. Hubbell, P. C., Digital Identification and Simulation of Nonlinear Systems with Physiological Applications, Ph.D. Dissertation, University of Pennsylvania, Philadelphia, Pennsylvania, 1967.

28. Diamessis, J. E., "A New Mthod for Determining the Parameters of Physical Systems, Proc. of the IEEE, V. 53, No. 2, p. 205-206, February 1965.

29. Diamessis, J. E., "On the Determination of the Parameters of Physical Systems," Proc. of the IEEE, V. 53, No. 3, p. 319-320, March 1965.

30. Wang, P. P., "On the Evaluation of the Parameters of a Nonlinear System," Proc. of the IEEE, V. 55, No. 3, p. 468-469, March 1967.

31. Golub, G., "Numerical Methods for Solving Linear Least Squares Problems," Numerische Mathematik, V. 7, p. 206-216, 1965.

32. Businger, P., and Golub, G., "Linear Least Squares Solutions by Householder Transformations," Numerische Mathematik, V. 7, p. 269-276, 1965.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Documentation Center Cameron Station Alexandria, Virginia 22314	2
2.	Library, Code 0212 Naval Postgraduate School Monterey, California 93940	2
3.	Professor George J. Thaler, Code 52Tr Department of Electrical Engineering Naval Postgraduate School Monterey, California 93940	2
4.	Mr. Joseph Rubis Naval Ships Research & Development Center Annapolis, Maryland 21402	1
5.	Mr. Hal Stephens International Business Machines Building 014 Monterey & Cottle Roads San Jose, California 95114	1
6.	Doctor Ken Deckert International Business Machines Monterey & Cottle Roads San Jose, California 95114	1
7.	ENS William R. Hansell USN 44 Ann Road Broomall, Pennsylvania 19008	1

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author) Naval Postgraduate School Monterey, California 93940	2a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED 2b. GROUP
3. REPORT TITLE IDENTIFICATION OF SYSTEM DYNAMICS USING MULTIPLE INTEGRATIONS	
4. DESCRIPTIVE NOTES (Type of report and, inclusive dates) Master's Thesis; June 1971	
5. AUTHOR(S) (First name, middle initial, last name) William R. Hansell	
6. REPORT DATE June 1971	7a. TOTAL NO. OF PAGES 102 7b. NO. OF REFS 32
8a. CONTRACT OR GRANT NO. b. PROJECT NO. c. d.	9a. ORIGINATOR'S REPORT NUMBER(S) 9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)
10. DISTRIBUTION STATEMENT Approved for public release; distribution unlimited.	
11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY Naval Postgraduate School Monterey, California 93940

13. ABSTRACT

A practical method for identifying linear time invariant systems on the basis of arbitrary input-output records is reviewed and extended to handle the case where the system is not initially in the zero state. The method is implemented using a digital computer program composed of a numerical integration subroutine and a subroutine for solving overdetermined sets of linear algebraic equations. Several examples are presented to demonstrate the accuracy and present capabilities of the procedure.

KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Linear Systems						
Identification						
Multiple Integrals						
Modeling						

Thesis
H1995 Hansell
c.1
Identification of system dynamics using multiple integrations.
129105